A FUZZY LOGIC BASED APPROACH FOR PARAMETRIC OPTIMIZATION OF ANALOG CIRCUITS

by

Biranchinath Sahu

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

December, 1999

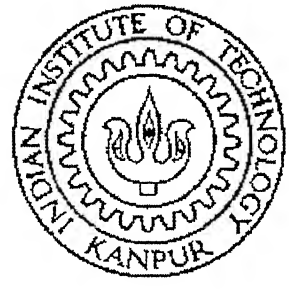
A FUZZY LOGIC BASED APPROACH FOR PARAMETRIC OPTIMIZATION OF ANALOG CIRCUITS

A thesis submitted in
partial fulfillment of the requirements
for the degree of

Master of Technology

By

Biranchinath Sahu

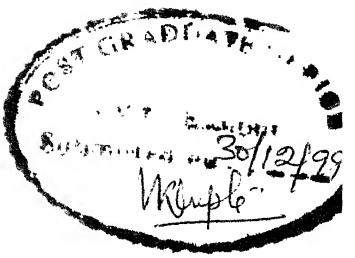


to the

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

December 1999.

CERTIFICATE



This is to certify that the work contained in the thesis entitled **"A Fuzzy Logic Based Approach for Parametric Optimization of Analog Circuits"** by **Biranchinath Sahu** has been carried out under my supervision and that this work has not been submitted elsewhere for the award of a degree.

30.12.99

(Dr. Aloke K. Dutta)
Associate Professor,
Department of Electrical Engineering,
Indian Institute of Technology, Kanpur.

December 1999.

ACKNOWLEDGEMENTS

I take this opportunity to express my sincere gratitude to Dr. Aloke K. Dutta for permitting me to work for my thesis under his guidance. I am thankful to him for his inspiration, guidance and the pain he has taken for correcting the thesis. It is his constant encouragement and motivation, made this work a success. Above all, it has been an exciting and a challenging experience to work under him.

I express my sincere regards to Dr. M. B. Patil, Dr. S. Kar, Dr. B. Mazhari, Dr. R. Sharan, Dr. Preeti Rao, and Dr. S. Umesh whose courses have helped me to raise my knowledge level and understanding. I would like to thank IIT Kanpur computer center for providing such an excellent computing facility.

Thanks to all the friends who have made my stay at IIT Kanpur a pleasurable, memorable and an enjoyable one.

Biranchinath Sahu

(Biranchinath Sahu)

ABSTRACT

The growing requirements for the single chip mixed-signal designs of very large scale integration (VLSI) together with the continuous trend towards smaller feature sizes and an even higher scale of integration have brought about new dimensions in the analog circuit design complexity. Because of the steady increase in the number of new application specific integrated circuit (ASIC) designs that include analog functions and their increasing complexity, the need for computer-aided design (CAD) tools is being felt. Over the years, a number of CAD tools have come into existence automating various parts of the analog design. The optimization approaches used for the design of analog circuits are found to be very much rigid in terms of capturing human intentions. In this work, we have used the concept of fuzzy membership functions in order to build a CAD tool for the parametric optimization of analog circuits. In order to capture human intentions in expressing the requirements for a particular application, e.g., *minimize power*, *maximize gain*, etc., for each of the performance specifications of a given topology, a membership function is assigned to each of them to measure the degree of fulfillment of the objectives and the constraints. A number of objectives are optimized simultaneously by assigning weights to each of them representing their relative importance, and then by clustering them to form the objective function, which is solved by Powell's direct search algorithm. Using this approach, some of the basic circuit topologies in bipolar and MOS technologies are optimized. The topologies considered for the bipolar circuits are the emitter-follower as an output stage, the common-emitter as an amplifier, and the common-base as an amplifier. A wide variety of MOS current sources (e.g., simple, cascode, Wilson, modified Wilson, and regulated cascode) and the common-source amplifier as a gain-stage have been optimized. The optimization routines for three basic CMOS op-amp topologies, e.g., the simple operational transconductance amplifier (OTA), the basic two-stage (BTS) op-amp, and the symmetrical OTA have been developed. The channel length modulation parameter (λ) is exclusively taken into account both in the DC operating point calculations and the small signal parameter computations for the MOS circuit topologies. Once the optimization is done, the program creates a SPICE netlist of the circuit topology for the verification of the design. The design results obtained from our optimization program showed an excellent match with those obtained from SPICE simulation for each of the topologies.

CONTENTS

LIST OF FIGURES

...

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Very large scale integration (VLSI) of devices in Integrated Circuits (IC) is now maturing with the current emphasis on deep submicron structures and sophisticated innovations, e.g., putting an entire system comprising of digital as well as analog circuits on a single chip, know as the system-on-a-chip (SOC). Today's advanced systems for telecommunications, robotics, automotive electronics, and image processing deal with extensive use of these types of applications. Exciting new applications are being unveiled in the field of neural computing, where massive use of analog/digital VLSI technologies will have significant impact [1]. To match the fast technological trend towards single chip analog/digital VLSI systems, effort has been put worldwide to produce advanced computer-aided tools for designing both digital and analog circuits. Architecture and circuit compilation, device sizing, and layout generation are a few familiar tasks on the world of digital IC design, which can be efficiently accomplished by matured computer-aided design (CAD) tools. In contrast, the development of tools for designing and producing analog ICs is still in a quite primitive stage and lacking the industrial penetration and acceptance already achieved by their digital counterparts. Therefore, the design time of mixed analog-digital systems is primarily dominated by the design time of the analog parts. The best way to design such mixed–signal systems rapidly is to develop CAD tools that can automatically design analog cells.

In digital design, the basic philosophy is that "*more is better*", i.e., packing density is the major concern among the IC manufactures with an aim to provide more number of functions in a chip, thus reducing the cost per function. The metal-oxide-semiconductor field-effect transistors (MOSFETs) are treated as simple switches and most of the design is

performed at a higher level of abstraction, i.e., at the system level, using hardware description languages (HDLs), and carrying out synthesis, verification, etc. of the designed circuit. Circuit improvements tend to be closely related to advances in fabrication technology; this allows the designer to work with the shortest possible channel length in order to obtain maximum possible packing density, and, at the same time, optimize the performance. In contrast, a good motto in analog design is that "*small is beautiful*". In analog designs, IC designers do not stress on packing density as much as in digital designs, rather it is the possible reduction in the size of the device, without hindering its behavior. The devices are not put too close to each other, considering the effects of the parasitic capacitances, which would deteriorate the performances of the circuit. The MOS devices are used in complex ways as designers make clever use of the subtle and detailed aspects of their behavior. As a result, the focus in circuit design is clearly on precision usage, and the circuit design is carried out at the transistor level itself. Device geometry is an essential part of the design: specific current requirements lead to computation of device sizes. Finally, it may be pointed out that the "*real world*" is inherently analog in nature, and square waves can not be transmitted through free space [2].

The problem of analog synthesis may be defined as the one of selecting the proper topology, sizing the circuit components, and laying out the structure in a manner that realizes the required functionality and meets the desired performance criteria. While the objective can be easily stated, the design of analog ICs that meet such criteria may be non-trivial. To appreciate the complexity of analog designs and their automation, it is worthwhile to compare analog and digital design mediums. The following conclusions may be derived from such a comparison.

1) In the digital domain, there are typically three main performance measures of interest, namely area, power, and delay. In contrast, performance parameters of analog circuits are numerous and depend on the functional block of interest.

2) The digital signal may be simply characterized as having two unique logic states. A consequence of these binary discrete levels is that the signal is subject to formal mathematical treatment using Boolean algebra. The analog signal, however, can carry

information in a variety of forms. The amplitude of incoming signal carries vital information in the case of an analog-to-digital converter. On the other hand, phase-locked loop systems process phase information and are indifferent to signal amplitudes.

3) In the digital domain, system performance may be readily expressed as linear function of subsystem performance with little loss of performance accuracy. In the analog domain, system performance is typically a nonlinear function of lower level attributes.

4) With the inclusion of the simple back propagation scheme, digital design methods such as standard cell and gate array permit layout to be considered independent of circuit topology selection and component sizing. In the analog domain, net parasitic capacitances can play a dominant role in determining attributes of high performance analog blocks. The use of analog standard cell libraries is not practical due to many divergent performance measures typical of analog systems. Critical layout parasitics must, therefore, be considered commensurate with topology selection and component sizing.

The advent and evolution of computers and digital signal processing methods resulted in a tremendous potential for processing power, which, in contrast to nature, could only process digital signals. Analog interface circuits have, therefore, become vital and indispensable parts for most of the digital circuits. They provide the necessary signal conditioning and modification such that they can be processed digitally. Interface circuits vary widely depending on specific functions and applications, such as data acquisition systems, A/D and D/A converters, particle and radiation detection circuits, automotive electronics, biomedical instrumentation and control circuitry, robot sensing, preamplifiers, power drivers, etc. [3]. During the last few years, a revival of industrial interest in analog ICs has resulted in a new series of circuit functions and higher levels of performance accuracy in the areas of microprocessor supervisory circuits, power supply ICs, battery back-up switches, massively parallel analog signal processors (neural networks), and switched-capacitor filters [4]. Analog circuits are the main signal processors in applications where area, power, and high frequency operation are the performances of concern, vastly outperforming their digital equivalents.

Economic and other factors favor the co-existence of analog circuits, working either as the interface blocks or the main signal processing circuits, and digital circuits on the same die. Application specific integrated circuits (ASICs) that are designed according to customer specifications, therefore, are moving steadily towards the integration of complete systems on a single-chip (SOC). It has been reported that in 1990, approximately 60% of all CMOS and BiCMOS ASICs were mixed analog and digital [1].

The growing requirements for single-chip mixed VLSI systems, together with the trend towards smaller feature sizes and higher scales of integration, have brought about new dimensions in the circuit design complexity. Digital designs have been completely automated with sophisticated CAD tools exploiting hierarchy and structured abstractions. In contrast, analog design is commonly perceived to be one of the most knowledge-intensive tasks and analog circuits are still designed largely by hand by experts, familiar with the tradeoffs involved in the performances and IC fabrication processes. The techniques needed to build good analog circuits seem to exist solely as expertise invested in individual designers. CAD tools specifically tailored to analog IC design promise to improve the design process in a variety of ways [4], which are detailed below.

1) *By shortening the design times:* This will considerably improve productivity. Moreover, the customer time-to-market requirements will be satisfied easily.

2) *By simplifying the design process:* As a consequence, more designers, starting from novices to experts, will be able to design standard analog circuits.

3) *By improving the likelihood of error-free designs from the first fabrication run:* Automating error-prone design tasks reduces the probability of making errors, and, therefore, decreases the design cycle/success ratio.

4) *By reducing the design and production cost:* This comes as a consequence of shorter design times and a smaller design cycle/success ratio.

5) *By improving the manufacturing yield:* Computer-aided methods can be used to estimate and enhance the manufacturing yield of circuits for different fabrication processes, thereby improving turnaround time and profitability.

6) *By allowing easier tracking of the fabrication processes:* CAD tools can be used to design circuits for different fabrication processes without additional efforts.

7) *By retaining the expert design knowledge:* The knowledge acquired by the design systems can be used for subsequent designs. It can be conveyed to novice users in the form of design examples, explanations of behavior, suggestions for modifications, conclusions, etc.

Analog circuit design is generally achieved through the following steps.

1. Topology selection.
2. Parametric optimization.
3. Layout generation.

The designer selects an appropriate topology among the various possible alternative architectures and topologies, in order to achieve higher performance for a particular application. The second step, after the topology of the circuit and component types are fixed, consists of assigning values to the circuit parameters (e.g., widths and lengths of MOS transistors, resistor and capacitor values, bias voltages and currents, etc.), while satisfying the desired performance criteria. The optimized circuit then need to be transformed into a layout.

1.2 LITERATURE REVIEW

Research in analog design automation (DA) has been relatively slow. By the year 1985, only a handful of analog DA systems were reported and only a few institutions worldwide demonstrated interest in analog DA. Many of these early systems were geared towards providing basic assistance to the designer and showed strong influence of the digital domain. It is only recently that the research interest in analog DA has been growing dramatically and a plethora of prototype systems, many of which are capable of handling full-custom designs, have been reported. A survey of the progress in the area of the analog DA can be found in the literature [5–7].

The distinction between CAD and DA tools is rather fuzzy in nature and a debatable one. A CAD tool is a computer-based system that provides assistance to a design task. This assistance can range from merely relieving a designer from long, tedious, and error-prone tasks of the design process to performing complete designs with minimal human intervention. The latter class of CAD tools that automate part or the whole of the design process is referred to as DA tools. DA tools that produce IC layouts straight from some form of high-level descriptions are also called silicon compilers [4]. Figure 1.2.1 depicts schematically a classification of the various analog circuit design approaches.

Layout-Based Design Approach

With the aim of providing a fast and reliable path to silicon, several systems have been developed that follow a layout-based design approach. In reality, this approach is an adaptation of extensively used standard cell, gate array, and parameterized cell methods found in the digital domain. Since with this approach, designs are controlled to a large extent by layout, it is also referred to as *semi-custom bottom-up approach* [4]. Analog arrays are pre-designed and laid-out blocks of different sizes, configurations, and levels of complexity, varying from single component arrays to circuit arrays. The required functions are designed by appropriately programming one or more levels of interconnect. Typical examples of circuit arrays are the SCA-6 and SCA-12 switched-capacitor filter arrays developed by Silicon Systems Inc. [8]. With these arrays, complex filters having more than 50 poles can be fabricated without requiring any external components [4]. This approach has the following two drawbacks.

1) They do not provide the necessary design flexibility required for high performance analog circuits. Not only there is a limited range of active and passive components available, but also their values are within a limited range. Thus, analog arrays can realize only a small number of discrete points within a vast and continuous performance space.

2) Arrays are not very much cost-effective in terms of silicon usage, since any unused components or circuits in the array simply waste silicon area without providing any function at all.

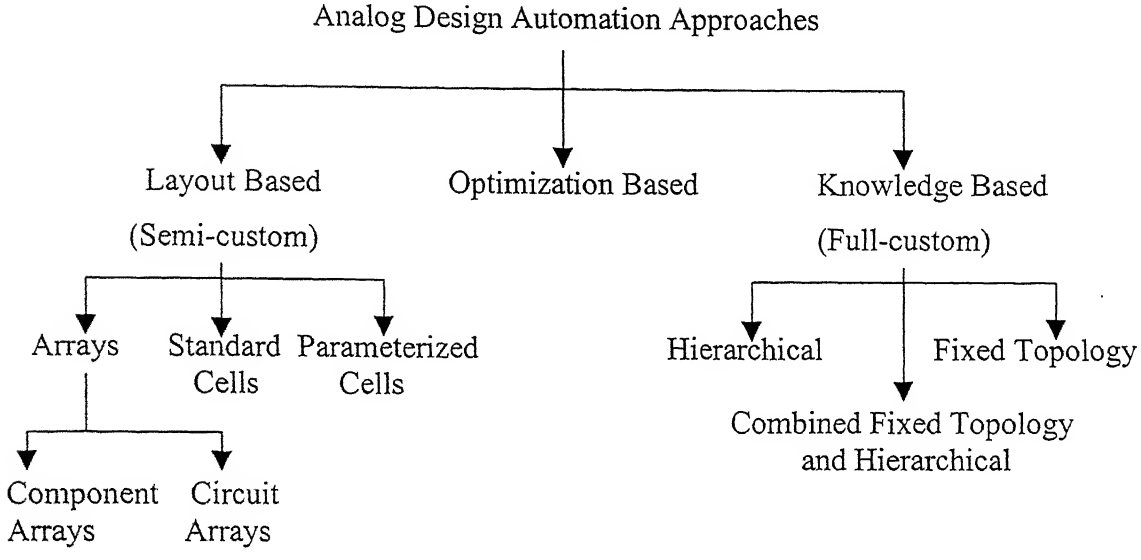


Fig.1.2.1: A classification of the various analog IC design automation approaches (taken from [4]).

Standard cells address the problem of silicon usage in a better fashion than the arrays. They are pre-designed and laid-out blocks of varying complexity that reside in the database of the DA system. The required function is implemented by assembling the necessary cells and then by placement and routing. Since only the required cells are fabricated, the chip area is not wasted. Although the use of standard cells may have been successful in the digital domain, it has a quite restrictive use for analog circuits, since it is extremely difficult to configure and maintain a rich enough library of cells in order to accommodate a wide spectrum of possible applications. Both arrays and standard cell libraries are very brittle, since they track the fabrication process very poorly. A number of CAD systems that use standard cell techniques are reported in the literature [9–11].

The use of analog parameterized cells is an alternative layout-based design approach. Parameterized cells are similar to the standard cells, but with some additional

flexibility gained by allowing some customization of the cells (or part of them) according to the required function. The degree of flexibility provided is directly dependent on the sophistication of the respective module generator – the piece of the code that generates the layout of the cell given a set of input parameters. With the application of this approach, AIDE2 has demonstrated the design of several circuits, e.g., amplifiers, integrators, switched-capacitor filters, and A/D converters [4].

Knowledge-Based Design Approach

Knowledge-based systems exploit domain knowledge in order to design analog ICs, and they address the design task in a full custom way, thereby allowing for maximum flexibility and a potentially better coverage of the circuits' performance space. So far, the main design philosophies that have evolved and prevailed are the *hierarchical* and the *fixed-topology* approaches, as shown in Fig.1.2.1. The third approach that has evolved from the first two, combines some features of both hierarchical and fixed-topology approaches.

Hierarchical Approach

The hierarchical design approach has been successfully applied to digital DA and is now seen in analog design also. The idea involves breaking of the required circuit (or system) into smaller distinct parts. Each of these parts is assigned a set of specifications which, if met, then the combination of these parts will yield the desired circuit performance. The process is repeated in a similar manner for smaller blocks at different hierarchical levels. The number of levels depends upon the complexity of the circuit as well as the sophistication of the design system. To be able to carry out such partitioning of circuits and decomposition of specifications, a huge amount of domain knowledge is required, and, generally, it is in the form of design equations and heuristics. These systems maintain the greatest degrees of freedom, and, thus, a small architecture library can lead to a large number of different topologies with wide performance spectra. The various systems that use this kind of approach are OASYS [12], BLADES [13], An_Com [14], etc.

Fixed Topology Approach

This method employs a sizing method in order to compute appropriate sizes for the devices within a given fixed circuit topology. These fixed, unsized, device level circuit topologies are stored in a knowledge base together with the necessary domain knowledge for dimensioning the devices. The nature of the domain knowledge depends on the method of computing the device sizes. Some of the systems reported in the literature that follow this approach are IDAC [15], OPASYN [16], OAC [17], etc.

Combined Hierarchical and Fixed Topology Approach

There are some knowledge-based design methods, which combine features of both the hierarchical and the fixed topology approaches. This approach is shown as a separate class in Fig.1.2.1. ASAIC [18] is a system that fits into this class of design systems, since it puts together the circuit topology in a hierarchical fashion, whereas the design of the individual device dimensions of the topology is performed in a manner that resembles those for fixed topology systems. It uses the symbolic simulator ISSAC [19] for the analytical modeling of the analog circuits. CAMP [20,21] is also one of such systems that designs a circuit first by viewing it as having a fixed topology, and, then, allows modifications to the various sections of the initial topology in order to meet the required performance specifications. Systems that combine features of both hierarchical and fixed topology approaches provide an additional degree of design flexibility in terms of topology modification. This can lead to a smaller circuit library and a wider coverage of circuit performances. ISAID [4,22] uses the concept of combined hierarchical and fixed topology approach, however, it also includes a circuit generator and a circuit corrector. The circuit generator is based on newly developed methods that are used to handle hierarchical generation of topologies. The circuit corrector is an application of qualitative reasoning, which analyses performance trade-offs without iterative simulation, thereby modifying topologies suitably. However, these are not as flexible as the hierarchical systems.

Optimization-Based Design Approach

The optimization-based design approach uses recent advances in the optimization theory and algorithms, and relates these to the parametric optimization of analog ICs. A survey of the various techniques used for optimization of ICs can be found in the literature [23,24]. The synthesis problem is formulated as one of mathematical programming. The circuit performances are considered to be the objective functions, which are to be minimized or maximized subject to a set of specification constraints. Optimization based design approaches can be broadly classified into two categories:

1. Simulation-based optimization.
2. Analytical equation based optimization.

Historically, the very first attempt towards analog DA were numerical optimization based. Systems such as DELIGHT.SPICE [25], ECSTACY [26], and the more recent ADOPT [27], consider the sizing of the individual transistors in a given circuit topology as an optimization problem. Typically, these systems employ optimization algorithms, which iteratively adjust the individual transistor sizes in order to meet the constraints and objectives specified by the user. A simulator is used within the optimization loop to assess the performance of the circuit during each iteration. These design approaches are referred to as the simulation based optimization. ASTRX/OBLX [28] developed at CMU also uses this technique. Systems based on numerical optimization techniques are independent of the actual circuit used, the technology, and the fabrication process. The use of these systems has been limited because of the following deficiencies.

1) The circuit designer has to specify a good starting point for the optimization algorithm. A bad starting point may lead to some local minimum, potentially rendering a good circuit useless. Simulated annealing [29] and random multi-start techniques, e.g., Genetic algorithms (GA) [30,31] attempt to overcome this problem, however, they are often computationally more expensive.

2) Specifying constraints and objectives as well as performance measuring procedures is usually a tedious process.

3) Generally, these systems are slow since they involve circuit simulation during each iteration within the optimization loop.

4) The circuit designer has to be familiar with not only the system itself, but also with the optimization algorithms used in order to either avoid or solve convergence problems and ensure satisfactory and reliable results.

5) The user has to have a certain amount of circuit design expertise in order to efficiently use such a system. Such design expertise is particularly important if the designer has to define the designable parameters that the optimizer has to vary in order to optimize circuit performance.

In order to avoid the time consuming and expensive simulator inside the optimization loop, several attempts have been made. One approach is to adopt simplified but sufficiently accurate analytical models that predict circuit performances, and these are used inside the optimization loop. This approach is referred to as the analytical equation based optimization. A number of prototypes have come out in recent times, which use this technique, e.g., OPASYN [16], STAIC [32], FPAD [33], FASY [34], and those reported by Mandal and Viswanathan [35] and Maulik et al. [36]. In the work reported by Taumazou and Markis [4], the problem of circuit modeling is simplified by decomposing it into equations describing circuit and device behavior. The design problem is formulated as that for a single objective optimization one, and solved by standard optimization techniques.

In OPASYN [16], the parametric optimization problem is formulated as that for an unconstrained optimization one, and is solved using the steepest descent algorithm [29]. The circuit performances are computed using analytical modeling equations. STAIC [32] features an input modeling language for entering hierarchical circuit descriptions, and a numeric solver unit that dynamically integrates analytical model equations. It is also

committed to find global optimal solution, and employs a successive solution refinement synthesis methodology.

In the work reported by Maulik et al. [36], all knowledge about the device behavior is embedded within the encapsulated device evaluator, which simplifies the description of the analog circuit that must be given by an expert designer. Multiple constraint optimization technique is used to solve the DC operating point with the Kirchoff's current law (KCL) and the Kirchoff's voltage law (KVL) constraints along with the performance constraints. The sequential quadratic programming (SQP) [37] is used to solve the optimization problem. The approach taken by Mandal and Viswanathan [35] is based on the observation that the first-order behavior of a MOS transistor in the saturation regime is such that the objective and the constraint functions can be modeled as having a convex nature. Second-order effects are handled by formulating the problem as one of those for solving a sequence of convex programs [38]. It claims to be capable of determining the global optimal solution to the problem for widely varying initial guesses.

1.3 OBJECTIVE OF THE PRESENT WORK

Analog circuit design is known to be a knowledge-intensive, multiphase, and iterative task that usually stretches over a significant period of time and is performed by designers with a large portfolio of skills. Therefore, it is considered by many as a form of art. This idea is further triggered by a lack of analog circuit design formalisms, i.e., neither there exists a circuit-independent design procedure for analog circuit design, nor does exist a formal representation, i.e., the equivalent of Boolean algebra in the digital domain, to relate the circuit function to its structure in a consistent way. The main obstacle to such developments is the very nature of analog signals that the circuit deals with, the continuous range of signal amplitudes, and their continuous time dependency.

Among the various sub-domains of analog design, i.e., topology selection, parametric optimization, and layout generation, each one plays an important role to achieve

higher performance for a particular application. Even if the topology of the circuit to be designed is fixed, finding the optimal set of circuit parameter values is not an easy and obvious task, since all the circuit performances are highly dependent on these choices. The parametric optimization is complicated by the conflicting design objectives and performance constraints, which are generally implicit nonlinear functions of the circuit parameters.

In all techniques used for parameter optimization, it is very crucial to formulate the problem in the right manner, and use an optimization algorithm that reflects the user's intentions as accurately as possible. These also are not easy and obvious tasks. Most of the optimization methods available are very rigid and often difficult to adapt to the design problems without a corresponding loss of accuracy. Consequently, most optimization techniques have the limitation of using rigid optimization problem that is too restrictive, thus eliminating or reducing the possibilities to arrange for trade-offs, which are important factors in the overall design process. As a result, the design space becomes limited, and, in most cases, no optimal solution can be found. Another problem is the choice of the starting point, on which the quality of the optimized solution and the design time heavily depend on. In most of the proposed techniques, this task is left to the user/designer.

The focal point of this work is aimed towards the development of a tool in order to find a set of circuit parameters (i.e., design variables), such that all the design objectives are optimized while satisfying performance constraints, assuming that the topology of the circuit to be designed is known. Also, an attempt is made to formulate the optimization problem in a realistic manner by capturing the human intentions in the inherently imprecise terms used by the user, e.g., *minimize* power, *small* output resistance, etc. The fuzzy set theory [39], as applied in FPAD [33], is used to formulate design objectives and constraints. Depending on the objectivity associated with a particular performance or constraint, it is transformed into either a fuzzy objective or a fuzzy constraint. The trade-offs among a number of specifications, depending on the users' interest, are handled by the means of membership functions. Each of the performance specifications is assigned a weight, which reflects its relative importance.

The performance functions of the circuit, as needed by the optimization routine, are evaluated by using analytical circuit equations, which describe the performance objectives in terms of the design variables. The technology parameters needed during the performance function computations are directly read from a technology file, making the synthesis routine technology independent[1]. The time consuming circuit simulator is avoided inside the optimization loop. The objective function is formulated by clubbing all the membership functions associated with the specifications in proportion to their assigned weights. Finally, the solution to the formulated problem is carried out using Powell's direct search algorithm [38,40]. The approach adopted in our work for synthesis of analog circuits combines the advantages of both the optimization-based and the knowledge-based approaches and tries to overcome the limitations of both of them.

This approach has the following advantages over the other DA approaches discussed earlier.

1. It uses more realistic formulation of the objective functions required for optimization, which supports the imprecision and vagueness inherently associated with the real-world formulation of any design problem. The formulation takes into account performance tolerances and allows varying degrees of a particular solution.

2. The approach adopted in this work is based on a constrained multi-objective algorithm, avoiding the limitations of both unconstrained and single objective optimization.

3. An automatic first-cut sizing procedure deduces the initial values for the various design variables for the optimization routine, based on the input specifications, design knowledge, and heuristics.

4. Analytic circuit equations and device models are used to avoid a simulator in the optimization loop, and, therefore, the cost of repeated simulation is avoided.

5. To allow trade-offs and limit repeated modifications of the input specifications, they are not assigned precise target values; instead, each specification is formulated as a

[1] This enables the user to use the same optimization routine for different technologies, e.g., 2 μm, 1 μm, 0.8 μm, etc., by only changing the technology file.

fuzzy set, i.e., a range of possible values are assigned to each of them with varying degrees of acceptability.

6. In particular, for MOS circuits, we have used a more accurate analysis, i.e., the channel length modulation parameter (λ) is taken into consideration, which had been neglected by most of the earlier approaches. The computation of the DC operating point is performed iteratively.

In order to explore the viability of the analog circuit design methodology as mentioned above, a CAD tool is constructed in this work in the C-language (having ANSI features) and applied for the synthesis of simple blocks in both bipolar and MOS technologies. The most widely used analog building block is the op-amp, which is almost an indispensable component of all analog systems. Synthesis of op-amps shows the nature of intricacies involved in the design of analog circuits; performances of op-amps track in different directions, and, therefore, optimization proves to be a challenge among analog circuit CAD tool developers. Realizing the immense importance of op-amps in analog circuits, we have applied the proposed approach for optimization for three different topologies (e.g., the simple operational transconductance amplifier (OTA), the symmetrical OTA, and the Miller compensated basic two-stage (BTS) op-amp). The bipolar circuits for which optimization modules have been developed in this work are the emitter-follower as an output stage, the common-emitter stage as an amplifier, and the common-base stage as an amplifier. Also, the synthesis procedures for a wide variety of current sources and a common-source amplifier as a gain stage in MOS technology have been automated in this work. The details of the performance specification and the synthesis procedure, along with the mathematical equations used for the design of the above mentioned analog building blocks are described in the subsequent chapters. The results obtained from the parametric optimization of different topologies are discussed along with the design routines in the respective chapters.

The development of the synthesis tool involved the creation of an optimization module in order to optimize an **n**-variable objective function. To provide the user a number of membership functions for transforming the performance specifications into fuzzy

objectives or constraints, a library is created. In order to develop the optimization modules for each of the topologies currently supported by our program, the following steps are followed.

- Mathematical conceptualization of the analog circuit topology, identification of the performance specifications and the design variables.
- Elimination of the unimportant and dependent design variables by using heuristics and circuit knowledge, thus filtering out the independent design variables, expressing the performance in terms of the independent design variables, and development of the computer code.
- Development of the program to read the performance specifications, type of constraints or objectives (e.g., *maximize*, *minimize*, etc.), weight of each of the performances, and, thereby, formulating the objective function for optimization.
- Development of the modules to create output files for storing the output results, operating point information, and an input file containing the netlist of the circuit topology for subsequent verification using SPICE simulation.
- Verification of the module by running it for a number of sets of performance specifications, obtaining the output, and their comparison with SPICE simulation.

The methodology adopted in this work can be interpreted as the development of a part of a complete analog circuit design tool. In other words, our attempt is to provide reliable and useful synthesis parametric optimization procedures for cell level analog circuit blocks in both bipolar and MOS technologies. An attempt has been made to look at the synthesis problem in a more systematic and organized way. The tool developed here can be used for evaluating the circuit parameters, considering the fact that almost all the important performances (a set of fourteen specifications) are considered for the design of op-amps. Also, the design results, as obtained from our program, match reasonably well with those obtained from SPICE simulations. While providing inputs to the tool, i.e., the performance specifications, tolerances, and weights, and the constant inputs such as supply

voltage, maximum and minimum values of the design variables, etc., it is assumed that the user has a good knowledge of the topology. The approximate values of the specifications that can be achieved with a circuit should be known to the user in order to avoid the possibility of providing absurd values as inputs. Otherwise, the tool may end up with an impractical result trying to satisfy all the objectives or may not produce an optimal solution at all, which is a problem for most optimization algorithms.

The thesis is organized in the following manner. Chapter 2 describes the overview of the methodology adopted in this work, i.e., the formulation of the problem, the concept of the fuzzy constraints, and the various types of membership functions. Chapter 3 presents the application of the proposed approach, used for designing simple BJT circuits, e.g., emitter-follower as an output stage, common-emitter stage as an amplifier, and common-base stage as an amplifier. Chapter 4 explains the design procedure used for simple MOS circuits, e.g., current sources (simple, Wilson, modified Wilson, cascode, and regulated cascode), and a common-source amplifier as a gain stage. Chapter 5 gives the detailed design aspects of the three types of CMOS op-amps for a given set of performance specifications and constraints, as mentioned earlier. The summary and conclusion along with the possible modifications and improvements needed to build a user-friendly tool using the approach adopted in this work have been discussed in Chapter 6.

CHAPTER 2

THE FUZZY OPTIMIZATION APPROACH

2.1 INTRODUCTION

This chapter deals with the explanation of the prototype DA program that has been developed in this work, and is applied for the synthesis of cell level analog circuits in both bipolar and MOS technologies. The general structure of the synthesis procedure adopted in this work can be represented by the means of a flow chart as shown in Fig.2.1.1.

The explanation of the synthesis algorithm goes as follows. First of all, the program asks the user to decide the circuit that he wants to optimize among the various topologies available. Once the circuit topology is decided, the program starts with a set of input performance specifications, such as the one given for op-amps in Table 2.1.1, and deduces the optimal set of design parameter values (e.g., capacitors and biasing resistors, lengths and widths of MOS transistors, bias currents, etc.). In the first step, the program interacts with the user to read the set of performance specifications for which the selected topology is to be optimized. Each of the performance specifications is either an *objective* or a *constraint*, which is to be specified by the designer. A *constraint* can either be a *fuzzy constraint* or a *strict constraint*. The *fuzzy constraints* are those which are expressed in real world terms, e.g., *maximize, minimize,* etc., with no precise target values, and these are the specifications for which the designer is allowed to have some flexibilities in terms of tolerances. On the other hand, *strict constraints* are those which should be met in a strict mathematical sense, e.g., output current of a current source; i.e., a current source is designed in order to get the output current as specified by the user, and its value can neither be maximized nor minimized.

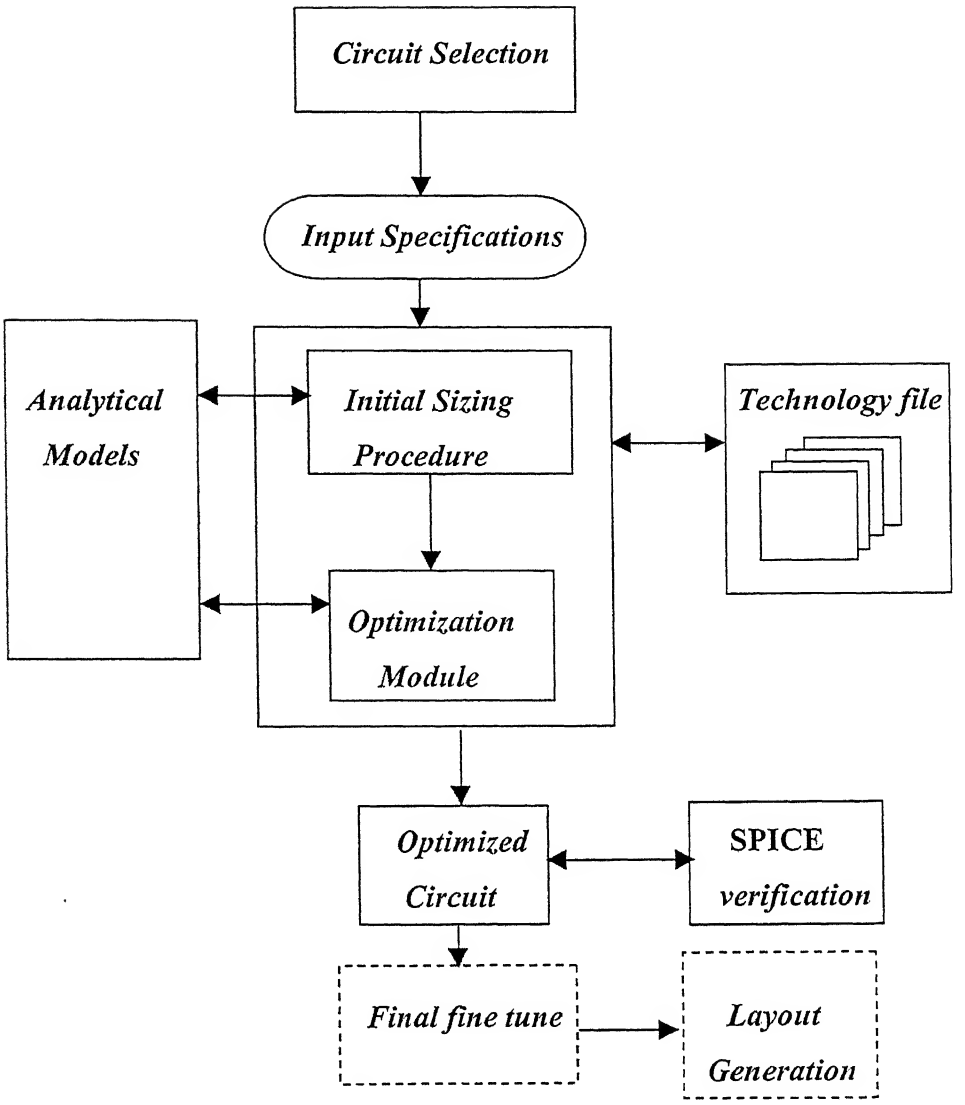


Fig.2.1.1: The general structure of the synthesis algorithm.

Each of the specifications and constraints is assigned a weight, which represents its relative importance in the set of objectives for which the selected topology is optimized. The sum of the weights assigned to each of the specifications and constraints is equal to unity. The details of the formulation of the objective function, which is subsequently optimized by a numerical optimization algorithm, is described in Subsection 2.2. A set of constant parameters, e.g., supply voltage, load capacitance, maximum length and width of transistors, etc., is supplied with the performance specifications as required by the program.

Table 2.1.1

Typical op-amp performance specifications for ± 5 V dual supply.

PARAMETERS	TYPICAL OBJECTIVE	TYPICAL VALUES
Open-loop gain	Maximize	≥ 70 dB
Phase margin	Maximize	≥ 45°
Slew rate	Maximize	≥ 10 V/μs
Unity-gain bandwidth	Maximize	≥ 3 MHz
Power dissipation	Minimize	≤ 10 mW
CMRR	Maximize	≥ 70 dB
PSRR	Maximize	≥ 70 dB
RMS noise	Minimize	≤ 1 μV/ √Hz
Output swing	Maximize	≥ ± 4 V

The second step is to obtain an initial solution for the objective function using automatic device sizing procedure, which is based on the circuit knowledge, basic assumptions, and first-order simplified analytical equations. This procedure implements expert design knowledge into procedural routines, which deduce device sizes from a set of input specifications. It may be noted here that the designer may choose to skip the initial sizing procedure and give his own initial guess as an input for the optimization module. The initial sizing procedures have been developed in this work for the design of operational amplifiers (op-amps) only, which are discussed in detail in Chapter 5. However, for the design of simple BJT and MOS circuits, the user is asked to provide the initial values of the design variables to be used for the optimization routine.

The fuzzy optimization module starts with the initial solution and identifies the optimal set of parameter values, which satisfy all the design objectives and constraints within a reasonable tolerance. However, if no solution exists for the given set of specifications, then the optimization module gives the result, which is the best among a set

of results for that particular case. Alternatively, it may be stated that the program finds a solution, which fulfils as many requirements specified by the user as possible. The program is aimed at arriving at a solution in the design space in order to obtain the designed values of the performances close to their corresponding specified values, which may not be fully met. After optimization, a file containing the netlist of the circuit topology is created by the program, which can be directly used as the input file for SPICE simulation in order to evaluate the performance of the designed circuit. The output results and the operating point information are stored in two separate files, and can be used for comparison with the results obtained from SPICE simulation.

During the optimization, all the technology dependent parameters, e.g., minimum channel length and threshold voltage of MOS transistors, current gain and early voltage of bipolar transistors, etc., are read from separate technology files, which contain all the parameters for a particular fabrication process. In order to optimize a circuit for another different technology, only the technology file needs to be changed, thus rendering this approach technology independent. Once the optimized circuit is obtained, it can be fine tuned using a simulation-based optimizer, such as DELIGHT.SPICE [25], in order to take into account the temperature and process variations. The final output can be taken in the layout form, in case the output is properly interfaced with a layout generation tool, e.g., ILAC as reported in IDAC [15], LASI [41], etc. The task of fine tuning and layout generation of the circuits are not implemented in this work, and, therefore, shown by dotted lines in Fig.2.1.1.

2.2 THE OPTIMIZATION MODULE

The optimization module, which deduces the values of the various design variables satisfying the specifications and constraints as required by the user, comprises of a number of submodules, and each of them is dedicated a particular task. The general structure of the module is represented as shown in Fig.2.2.1.

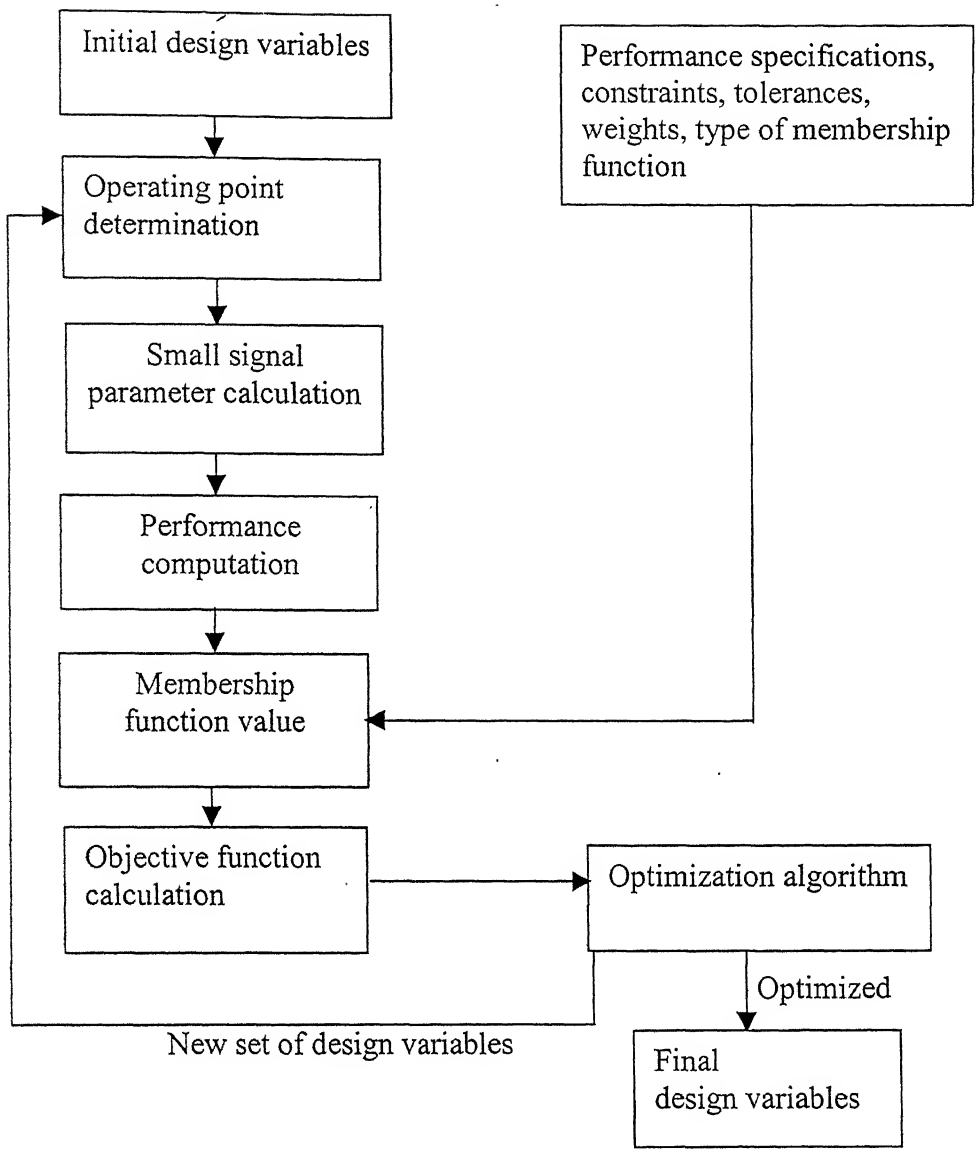


Fig.2.2.1.A flow chart representation of the optimization module developed in this work.

The initial design variables are obtained from the initial sizing procedure for the op-amp circuits, whereas for bipolar circuits and MOS current mirrors, the user is asked to provide the initial guess. The operating point is determined using these initial values of the design variables and the constant parameters, which are supplied by the designer. The procedure for the computation of the operating point is not a generalized one, and is very much dependant on the particular type of circuit topology. The MOS circuits require

iterative solutions in order to evaluate the different node voltages, when the channel length modulation parameter (λ) is taken into consideration.

After the operating point is determined, the small signal parameters are calculated using the relevant mathematical equations. In the next step, the performances are calculated using analytical equations. The mathematical equations used for all the computational work are mentioned along with the description of the circuit topology in subsequent chapters. The value of the *membership function* assigned to a particular performance specification is determined from its computed value, the target value to be achieved, and the tolerance limit.

After the values of all the membership functions are computed, they are clubbed together in order to form a single objective function in proportion to their assigned weights, which is solved by Powell's optimization algorithm [38]. During the optimization process, if the objective function reaches either the value of unity or a maximum, the program terminates, and the output files are created. Otherwise, the algorithm changes the design variables suitably in order to meet the objective function in the best possible way. The details of the algorithm used for optimization is described in Appendix–I.

2.3 FORMULATION OF THE OBJECTIVE FUNCTION

The formulation of a design problem as a standard mathematical programming[1] one is not always an obvious and easy task, because intensive knowledge about the performance specifications and design parameters is required. Unconstrained optimization is not suitable for synthesis of ICs, since there is always some constraints on the performance specifications and the design parameters. Also, the suitability of the single objective optimization approach, which optimizes a particular specification at a time, is of a restrictive nature, since it asks the designer to decide which performance specification

[1] To apply optimization algorithm for design of circuits, a mapping of the design problem into a mathematical function is required, which can be solved using the optimization technique.

among a set is to be optimized. On the other hand, since our requirement is to optimize several specifications at the same time, hence, the multiple objective optimization approach is adopted in this work in order to solve the above problem.

The design problem can be formulated as a mathematical programming problem in the following manner

$$\begin{aligned} &\text{Maximize or minimize} \quad \{f_1(x), \ldots\ldots\ldots, f_m(x)\}, \\ &\text{subject to:} \quad g_i(x) \leq \text{or} \geq \text{or} = spec_i \qquad i = 1, 2, \ldots, n, \\ &\text{and} \quad x_{min} \leq x \leq x_{max}, \end{aligned} \tag{2.1}$$

where $f_j(x)$ $[j=1, 2, \ldots, m]$ are m objective functions which are to be minimized or maximized, $g_i(x) \leq$ or $\geq$ or $=$ $spec_i$ are n constraints which are to be satisfied, $spec_i$ is the limiting value of the i^{th} specification, x is the vector of the design parameters (e.g., lengths and widths of the various transistors, values of resistors and capacitors, bias voltages and currents, etc.), and x_{min} and x_{max} are the minimum and the maximum values of the design parameters respectively.

2.4 FUZZY OBJECTIVES AND CONSTRAINTS

During the formulation of the design problem as a mathematical programming function, the requirements of the designer are not always well defined. More elaborately, it can be stated that there is no definite method as to how to choose the various objective functions and what precise values are to be assigned to each specification, as given by Eqn.(2.1). Also, the designer may not be well aware of what may be the best design, because it depends on the variety of performance specifications for which the given circuit topology can be designed, i.e., for a given set of performance specifications, there may exist a better design than that obtained. To model the design problem as given by Eqn.(2.1), the designer is forced to define his problem in strict mathematical terms rather than in real world terms, which are more diversified in nature. In other words, it can be

explained that the real world terms, e.g., *small area, minimize power dissipation*, etc., do not indicate any precise numerical values to be used by the various optimization algorithms. In fact, the designers' objectives and constraints can be better stated in real world terms rather than by precise numbers. The designers often use terms, e.g., *minimize, small, very large, substantially higher than*, etc., in order to state their objectives and constraints for a given design problem. However, these terms possess fuzzy meanings, and, therefore, are difficult to express by precise values required for the problem formulation as per Eqn.(2.1). It is possible to quantify and manipulate such human statements using fuzzy set theory [39].

For this reason, with each objective function $f_i(x)$ in Eqn.(2.1), a fuzzy set is associated that formulates the fuzzy meaning of each of the performance objectives and what the designer actually wants to achieve with that. In the same manner, the terms used by the designer, e.g., *high* gain, *small* output resistance, *maximize* slew rate, *minimize* area and power, *maximize* bandwidth, etc., can be formulated using fuzzy sets and treated as fuzzy objectives. While attempting to minimize a performance function, designers often stop the search procedure when it attains a minimum value, even though that minimum may be local instead of global. Additional searching may be extremely time consuming with no apparent improvement in the objective function. In order to overcome the above problems, with each objective function given in Eqn.(2.1), one fuzzy set is associated, which formulates the fuzzy meaning of *minimize (maximize),* and, thereby, the intention of the designer is expressed more precisely. In the same manner, the performance specifications stated in real world terms, e.g., *high* gain, *small* output resistance, etc., are formulated using fuzzy sets and are treated as fuzzy objectives.

For each fuzzy objective, a *membership function* is defined, which associates a grade of membership $\mu_{fi}(x)$ with each $f_i(x)$ of the objective function, which reflects the degree of acceptability of that particular performance value. If D_{fi} is the interval of the possible values of $f_i(x)$, then μ_{fi} is defined as

$$\mu_{fi}: \quad D_{fi} \rightarrow [0, 1]$$
$$f_i(x) \rightarrow \mu_{fi}(x), \tag{2.2}$$

where $\mu_{fi}(x)$ is a real number in the interval [0, 1], reflecting the degree of fulfillment of the fuzzy objective associated with the objective function $f_i(x)$. More clearly, it can be stated as: $\mu_{fi}(x) = 1$ means that $f_i(x)$ is fully satisfied; on the other hand, if $\mu_{fi}(x) = 0$, then $f_i(x)$ is not satisfied at all, which means that it takes a value that is totally unacceptable to the designer. An intermediate value of $\mu_{fi}(x)$ between zero and unity reflects the acceptability of that particular performance value. It is quite obvious that the closer the value of $\mu_{fi}(x)$ is to unity, the better is the solution. An example of the membership function for the fuzzy objective *minimize power dissipation Pdis(x)* is shown in the Fig.2.4.1, which shows two plots of the acceptability function with respect to the variation of power dissipation *Pdis(x),* with one corresponding to the non-fuzzy representation and the other corresponding to the fuzzy membership function. In the non-fuzzy representation, the performance specification is accepted by the user only when it is satisfied strictly in mathematical sense, whereas the fuzzy membership function permits some value of acceptability between zero and unity depending on the level of its match with the specification. For the non-fuzzy case, the value of acceptability is equal to unity only when *Pdis (x)* is less than 10 mW. On the other hand, in fuzzy representation, there exists a value representing the degree of acceptability even if *Pdis (x)* is greater than 10 mW. It is quite obvious from the plot that in the fuzzy representation, an effort is made to reduce the power dissipation close to zero, a characteristic which does not exist in the non-fuzzy type of representation. In this work, different types of membership functions are used which can be assigned to the various performance specifications for the formulation of the objective function, the details being described in Subsection 2.5.

The formulation of the problem as per Eqn.(2.1) can thus be replaced as follows:

$$\begin{aligned} &\text{Maximize: } \{\mu_{f1}, \mu_{f2}, \ldots\ldots\ldots\ldots, \mu_{fm}\} \\ &\text{subject to: } g_i(x) \leq \text{ or } \geq \text{ or } = spec_i \qquad i = 1, 2, \ldots, n, \\ &\text{and} \qquad x_{min} \leq x \leq x_{max}. \end{aligned} \tag{2.3}$$

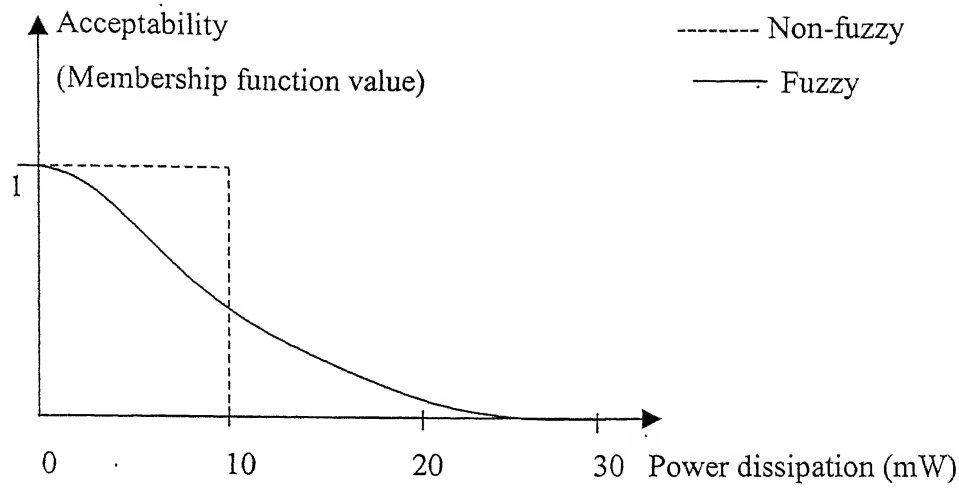


Fig.2.4.1: Membership function for the fuzzy objective *minimize power dissipation Pdis(x).*

Obtaining the design vector x that fully satisfies Eqn.(2.3) is generally not a trivial task, since the design objectives are often conflicting in nature. In that case, one has to yield to some compromises by permitting some tolerances in the inequalities. Thus, the symbol "$\leq$" in Eqn.(2.3) is replaced by the fuzzy version of "$\leq$", which means *essentially smaller than* [39]. Similarly "$\geq$" and "=" are replaced by their fuzzy versions *essentially greater than* and *essentially equal to* respectively. This formulation allows some violation in the constraint and provides a measure of this violation. Thus, the constraint $g_i(x) \leq$ or $\geq$ or $= spec_i$ becomes a *fuzzy constraint.* Each fuzzy constraint is characterized by a membership function $\mu_{gi}(x)$, which reflects the degree of fulfillment of the fuzzy constraint $g_i(x) \leq$ or $\geq$ or $= spec_i$.

During the process of defining the membership functions for the objective functions $[\mu_{fi}(x)]$ and the performance constraints $[\mu_{gi}(x)]$, it is essential that the difference between *fuzzy objectives* and *fuzzy constraints* be taken care of. For a *fuzzy objective*, the membership function must reflect a constant effort to improve the corresponding performance function $f_i(x)$. On the other hand, in case of a *fuzzy constraint*, once the threshold is achieved, no further effort is made to improve the performance function.

Figure 2.4.2 shows a comparison of the *fuzzy objective* and the *fuzzy constraint*, when associated with the performance specification *Gain > 40 dB*.

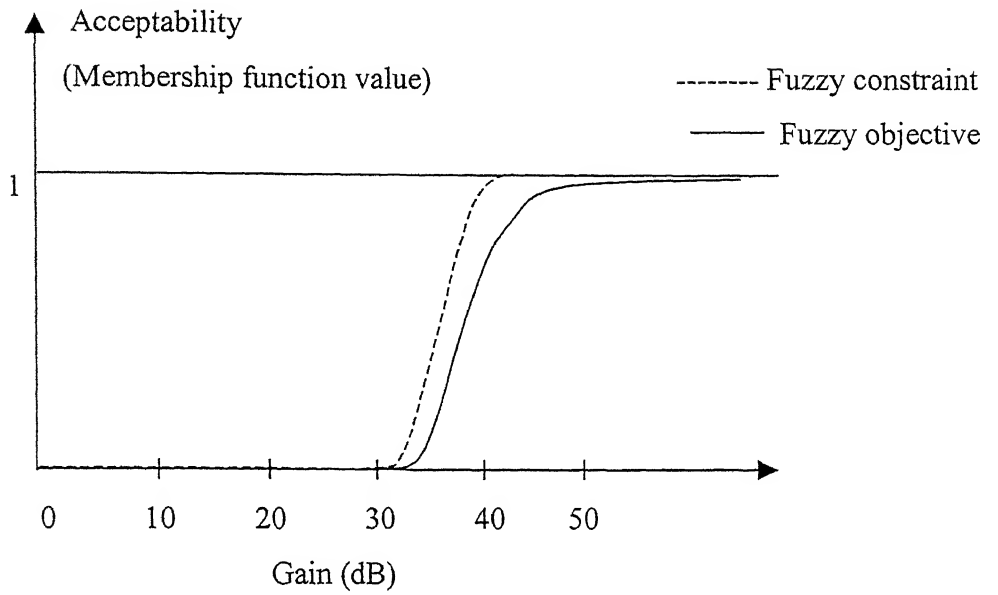


Fig. 2.4.2: A comparison of the membership functions for the *fuzzy objective* and the *fuzzy constraint* for the specification *Gain > 40 dB*.

It is quite obvious from Fig.2.4.2 that when the specification *Gain > 40 dB* is modeled as a *fuzzy constraint*, with the objective function being one of the various specifications, e.g., *minimize area, minimize power dissipation*, etc., the value of the membership function improves only up to the point where the specification is fully met. On the other hand, when it is modeled as a *fuzzy objective*, the membership function value, while continuously increasing itself towards unity, pulls the value of the gain beyond the specified target value. Thus, the membership function value reaches unity at a very high value of gain (may be 80 dB or so in this case), and, thereby, an effort is put to maximize the performance specification.

After the objectives and the constraints are fuzzified, i.e., the corresponding membership functions are defined, the design formulation of Eqn.(2.3) becomes:

$$\text{Maximize: } \{\mu_{f1}, \mu_{f2}, \ldots\ldots\ldots \mu_{fm}, \mu_{g1}, \mu_{g2}, \ldots\ldots\ldots\ldots \mu_{gn}\}$$
$$\text{subject to: } x_{min} \leq x \leq x_{max}. \tag{2.4}$$

In order to solve the above problem, there is a need to compose the membership functions μ_{fi} and μ_{gi} into a single synthesis membership function $\mu_D(x)$ by applying some algebraic operators. One possible method is to assign weights to each membership function, thereby giving a relative importance to each of them [33]. This has been implemented in our approach in order to club the membership functions together. Mathematically, it can be given by

$$\mu_D(x) = \sum_{i=1}^{m} w_i \mu_{fi} + \sum_{i=1}^{n} w_i \mu_{gi}, \tag{2.5}$$

where w_i and w_j are the weights giving the relative importance of each of the objectives and the constraints. The single synthesis membership function $\mu_D(x)$ can be thought of as a global design quality measure or a figure of merit for a particular design as a function of the design parameters x. Ideally $\mu_D(x) = 1$, which indicates that all the design objectives and constraints are met. On the other hand, $\mu_D(x) = 0$ indicates that there is no feasible solution for the design specifications.

Finally, the formulation of the design problem becomes:

$$\text{Maximize: } \mu_D(x)$$
$$\text{subject to: } x_{min} \leq x \leq x_{max}. \tag{2.6}$$

This formulation has two advantages over Eqn.(2.1). Since the maximum value of $\mu_D(x)$ is known to be equal to unity, hence, each iteration gives an idea about how far one is from the optimal solution. Also, it interacts more with the user, since one can manipulate the shapes of the membership functions and their compositions, as illustrated in the next section.

2.5 MEMBERSHIP FUNCTIONS

The prototype design automation program developed in this work asks the designer to assign membership functions for each of the performance objectives and constraints. Different types of membership functions are assigned to the various objectives depending on their nature, e.g., *minimize*, *maximize*, etc. The prototype program currently takes care of five types of membership functions, the details of which are described later in this subsection. The purpose of assigning membership functions to the performance specifications is to assign a mathematical meaning to the real world terms in order to capture the intention of the user, and, thereby, having a strong hold over the assignment.

Considering the various types of objectives and constraints, the membership functions can be broadly categorized into three items, i.e., ***greater than equal (≥), less than equal (≤), and equal to (=)***. Each category of the membership functions can be implemented in five different ways, thereby different profiles of the variations can be obtained. The mathematical expressions used for the formulation of a particular type of membership function, e.g., ***less than equal,*** in all the five different ways are presented here as an example.

2.5.1 Membership Function No. 1:

$$\mu_{gi}(x) = \exp\left\{-\left(\frac{g_j(x) - spec_j}{p_j / 2.0}\right)^2\right\}, \qquad g_j(x) > spec_j$$

$$= 1. \qquad g_j(x) \le spec_j \qquad (2.5.1)$$

2.5.2 Membership Function No. 2:

$$\mu_{gi}(x) = \exp\left\{-\left(\frac{g_j(x) - spec_j}{p_j / 2.0}\right)\right\}, \qquad g_j(x) > spec_j$$

$$= 1. \qquad g_j(x) \le spec_j \qquad (2.5.2)$$

2.5.3 Membership Function No. 3:

$$\mu_{gi}(x) = 1 - \left(\frac{g_j(x) - spec_j}{p_j} \right), \quad spec_j \le g_j(x) \le spec_j + p_j$$
$$= 1. \qquad g_j(x) \le spec_j$$
$$= 0. \qquad g_j(x) \ge spec_j + p_j \qquad (2.5.3)$$

2.5.4 Membership Function No. 4:

$$\mu_{gi}(x) = \frac{\left\{ 1 + \cos\left(\pi \left(\frac{g_j(x) - spec_j}{p_j} \right) \right) \right\}}{2.0}, \quad spec_j \le g_j(x) \le spec_j + p_j$$
$$= 1. \qquad g_j(x) \le spec_j$$
$$= 0. \qquad g_j(x) \ge spec_j + p_j \qquad (2.5.4)$$

2.5.5 Membership Function No. 5:

$$\mu_{gi}(x) = \frac{1}{\left(1 + \frac{\left(g_j(x) - spec_j \right)^2}{p_j} \right)}, \qquad g_j(x) > spec_j$$
$$= 1. \qquad g_j(x) \le spec_j + p_j \qquad (2.5.5)$$

Figure 2.5.1 shows the shapes of the membership functions as expressed by Eqns.(2.5.1–2.5.5). Each of the membership function characterizes the given fuzzy inequality constraint $g_i(x) \le spec_j$, $p_j > 0$ is the tolerance interval, i.e., the range of the degrees of acceptability, and $spec_j$ is the specified value of the performance to be met. The mathematical equations for the other two types of membership functions ***greater than equal*** and ***equal to*** are not mentioned here for brevity. However, their shapes for different representations similar to the ones given above are shown in Figs.2.5.2 and 2.5.3 respectively.

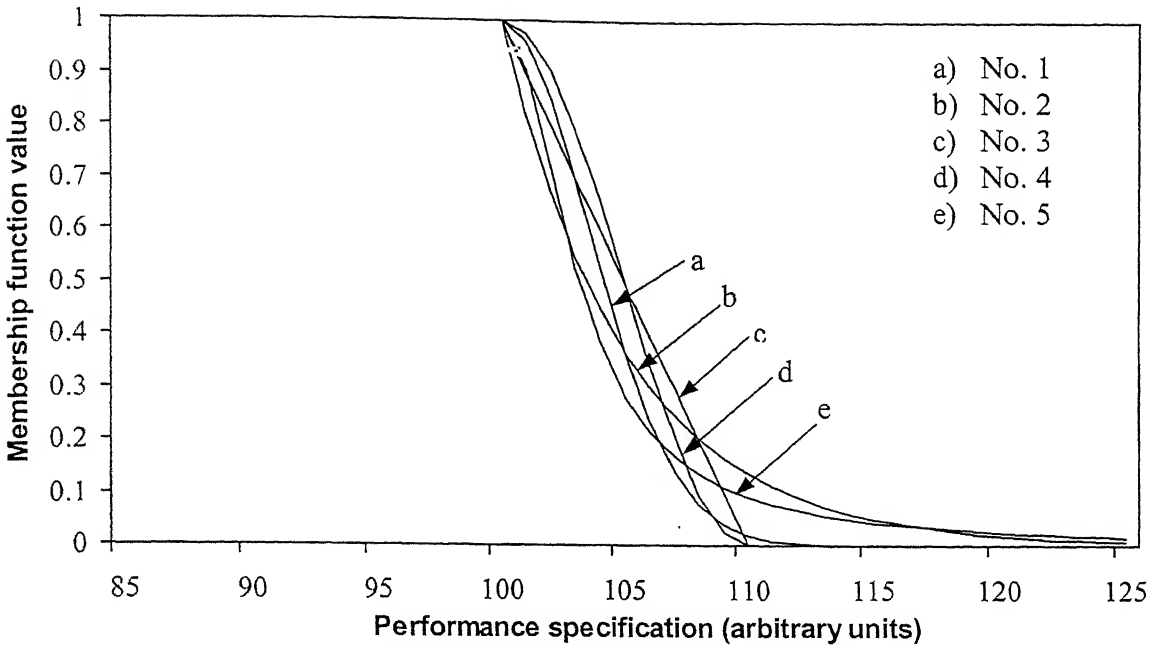


Fig. 2.5.1: The different shapes of the membership function ***less than equal***, which can be used for the constraint $g_i(x) \leq spec_j$ ($spec_j = 100$, $p_j = 10$) as implemented by our program. No.1 to 5 correspond to Eqns. (2.5:1) to (2.5.5) respectively.

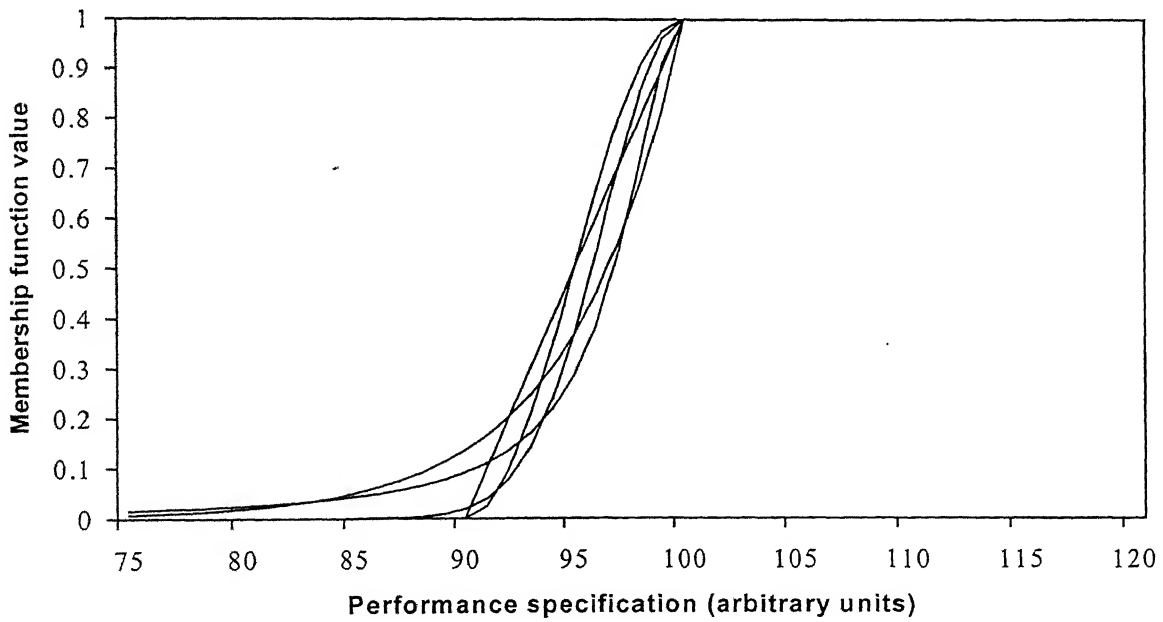


Fig. 2.5.2: The different shapes of the membership function ***greater than equal***, which can be used for the constraint $g_i(x) \geq spec_j$ ($spec_j = 100$, $p_j = 10$) as implemented by our program.

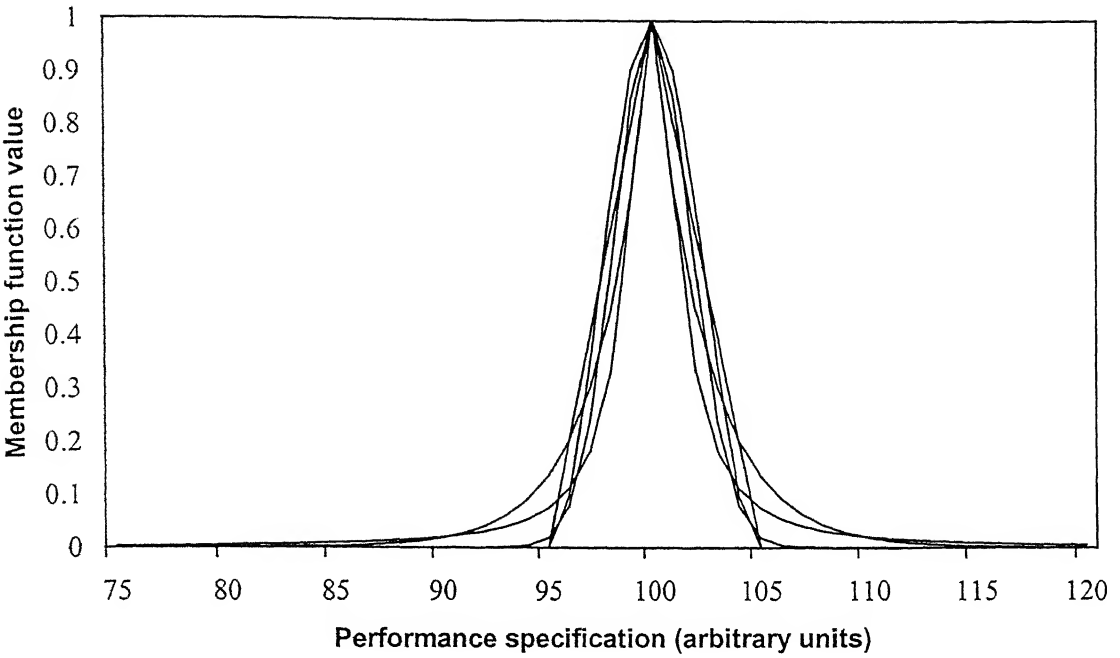


Fig. 2.5.3: The different shapes of the membership function ***equal to***, which can be used for the equality constraint $g_i(x) = spec_j$ ($spec_j = 100$, $p_j = 10$) as implemented by our program.

CHAPTER 3

OPTIMIZATION OF SIMPLE BJT CIRCUITS

3.1 INTRODUCTION

In order to examine the validity of the approach proposed in the previous chapter, an attempt has been made to optimize simple circuits using bipolar junction transistors (BJTs). The technology parameters, e.g., current gain (β_F and β_0), Early voltage (V_A), etc., are taken from the technology file for a 5 Ω-cm, 17 μ epi, 44 V *npn*-device process [42], a summary of which is given in Appendix–II. In this work, three basic circuit topologies, e.g., the emitter-follower as an output stage, the common-emitter stage as an amplifier, and the common-base stage as an amplifier using *npn* transistors are separately taken into consideration, and then optimized with respect to their performance objectives. The synthesis procedures for these three cases are explained in the subsequent subsections.

3.2 THE EMITTER-FOLLOWER AS AN OUTPUT STAGE

The schematic of an emitter-follower as an output stage is shown in Fig.3.2.1. This is one of the simplest analog circuits that has been considered for optimization in our program. In the first step, the program asks the user for the performance specifications, which are to be attained for the given topology. In this case, the circuit has been designed to meet the performance specifications presented in Table 3.2.1 along with the typical objectives for each parameter. In Table 3.2.1, the terms maximize, minimize, and threshold can be differentiated from the variation of the membership functions associated with the performance specifications during the optimization process. In case of maximization or minimization of the performance objectives, the corresponding membership function is formulated in such a way that it drives the objective to either increase or decrease. On the

other hand, the specifications having objective as threshold, once the specified performance value is achieved, the membership function maintains the value of unity, and, thereafter, no further effort is put to either increase or decrease their values.

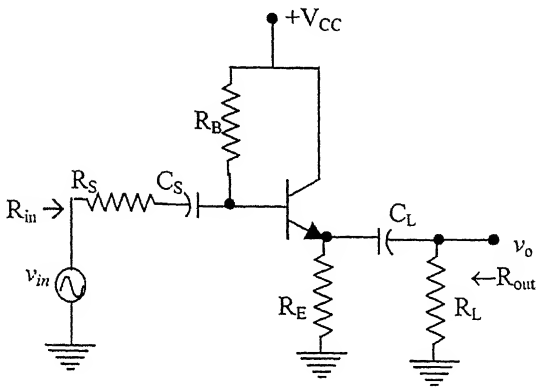


Fig.3.2.1: Schematic of an emitter-follower as an output stage.

Table 3.2.1

The set of performance specifications for the emitter-follower as an output stage as implemented in our program.

Sl. No.	*Symbol*	*Specification*	*Typical objective*
1.	V_{opp}	Output peak-to-peak voltage	Maximize
2.	*delV*	Clipping of the output voltage	Minimize
3.	R_{out}	Output resistance	Minimize/Threshold
4.	R_{in}	Input resistance	Maximize/Threshold

In the second step, the user is asked to specify the membership function for each of the performance specifications, choosing one among the various types available, as described earlier in Subsection 2.5. For the formulation of the objective function, the user is asked to provide the target that he wants for each of the performance parameters, and the tolerances that he can have for each of them. Also, the user has to assign weights to each of the membership functions, such that the sum of the weights equals unity. Unless specified,

the program assigns equal weights for each of the performance specifications. Taking these as inputs, the overall membership function is formulated.

The design variables in this case are the two biasing resistors R_E and R_B, and the coupling capacitors C_S and C_L. However, only the biasing resistors R_E and R_B are considered to be independent design variables. The values of the coupling capacitors are obtained from the consideration of the lower-cutoff frequency. The algorithm used for calculating the values of these capacitors is described later in this section. The source resistance R_S, the load resistance R_L, and the supply voltage V_{CC} are taken by the program as constant inputs. Using these, the DC operating point, the small signal parameters, and the performance parameters are calculated from the mathematical expressions given below.

1. The base current (I_B) is given by

$$I_B = \frac{V_{CC} - V_{BE}}{R_B + (\beta_F + 1)R_E}, \tag{3.2.1}$$

where V_{BE} is the base-emitter voltage (typically 0.7 V for silicon transistors), and β_F is the DC short-circuit common-emitter current gain of the transistor.

2. The collector current (I_C) is given by

$$I_C = \beta_F I_B. \tag{3.2.2}$$

3. The transconductance (g_m) is given by

$$g_m = \frac{I_C}{V_T}, \tag{3.2.3}$$

where V_T is the thermal voltage (= kT/q, where k is the Boltzmann constant, T is the temperature in K, and q is the electronic charge).

4. The input resistance (r_π) of the transistor is given by

$$r_\pi = \frac{\beta_0}{g_m}, \tag{3.2.4}$$

where β_0 is the ac small signal current gain of the transistor. Typical values of β_0 are close to those of β_F [42], and, therefore, a single value of β is used for both the DC and the ac calculations in this work.

5. The collector-to-emitter quiescent operating voltage(V_{CEQ}) is given by

$$V_{CEQ} = V_{CC} - (\beta_F + 1)I_B R_E. \tag{3.2.5}$$

In order to define the performance parameters output peak-to-peak voltage (V_{opp}) and output voltage clipping ($delV$), a typical load-line characteristic of the stage is taken for consideration as shown in Fig.3.2.2. During the positive half-cycle of the applied ac signal to the input of the emitter-follower, the base current increases and the collector-to-emitter voltage decreases, which can reduce up to the voltage V_{CESAT} (the collector-to-emitter saturation voltage, a technology parameter), beyond which the BJT enters the saturation region. Therefore, the maximum value of the output voltage ($V_{o\max}$) can be given by the following expression

$$V_{o\max} = V_{CC} - V_{CESAT}. \tag{3.2.6}$$

The positive swing of the output voltage is towards V_{CC} with respect to the quiescent point (Q-point) output voltage $V_{OUT,DC}$ ($= V_{CC} - V_{CEQ}$), the maximum value of which is defined as V^+, and it can be given by

$$V^+ = V_{o\max} - V_{OUT,DC} = V_{CEQ} - V_{CESAT}. \tag{3.2.7}$$

Alternatively, during the negative half of the ac signal, the output voltage can swing up to the minimum voltage at point A as shown in Fig.3.2.2, below which the transistor enters the cutoff region. At this point, the minimum output voltage ($V_{o\min}$) is given by

$$V_{o\min} = V_{CC} - V_{CEQ} - I_E(R_E \| R_L). \tag{3.2.8}$$

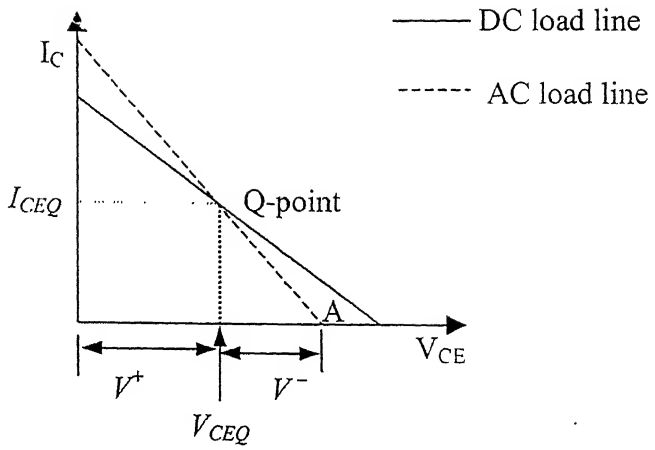


Fig.3.2.2: A typical load line characteristic of the emitter-follower stage.

The negative swing of the output voltage is towards the ground potential with respect to the DC output voltage $V_{OUT,DC}$ ($= V_{CC} - V_{CEQ}$), the maximum value of which is defined as V^-, and it can be given by

$$V^- = V_{OUT,DC} - V_{o\min} = I_E(R_E \parallel R_L). \tag{3.2.9}$$

The output peak-to-peak voltage (V_{opp}) is defined as the possible voltage swing between the instantaneous maximum and minimum values of the output voltage without the transistor entering either the saturation or the cutoff region. Mathematically, it can be given by the following expression

$$V_{opp} = V_{o\max} - V_{o\min} = V^+ + V^- = V_{CEQ} - V_{CESAT} + I_E(R_E \parallel R_L). \tag{3.2.10}$$

In order to get a symmetrical output voltage, the Q-point must be chosen such that the voltage swing on either side of it is the same. However, if the positive and the negative swings are not the same, then the output voltage gets clipped. The amount of clipping ($delV$) is obtained from the difference between the positive and the negative swings, and can be given by

$$delV = V^+ - V^- = V_{CEQ} - V_{CESAT} + I_E(R_E \parallel R_L). \tag{3.2.11}$$

The input resistance (R_{in}) of the circuit is given by

$$R_{in} = R_S + (R_B \,||\, R_x), \tag{3.2.12}$$

where $R_x = r_\pi + (\beta_0 + 1)(R_E \,||\, R_L).$

The output resistance (R_{out}) of the circuit is given by

$$R_{out} = R_E \,||\, R_Y \,||\, R_L, \tag{3.2.13}$$

where $R_y = (r_\pi + R_S \,||\, R_B)/(\beta_0 + 1).$

The values of the coupling capacitors C_S and C_L are calculated from the values for the biasing resistors obtained from the optimization routine, satisfying all the performance specifications as specified by the user, and the consideration of the lower cutoff frequency using the following algorithm. The expression for C_S is given by

$$C_S = \frac{1}{2\pi R_{CS} f_{LS}}, \tag{3.2.14}$$

where f_{LS} is the lower cutoff frequency due to C_S, and R_{CS} is the resistance seen by C_S, and is the same as R_{in}, given by Eqn.(3.2.12). The expression for C_L is given by

$$C_L = \frac{1}{2\pi R_{CL} f_{LL}}, \tag{3.2.15}$$

where f_{LL} is the lower cutoff frequency due to C_L, and R_{CL} is the resistance seen by C_L, given by

$$R_{CL} = R_L + R_E \,||\, R_Y. \tag{3.2.16}$$

The lower cutoff frequency (f_L) of the circuit is attributed due to both f_{LL} and f_{LS}, and can be given by the following expression

$$f_L = \sqrt{f_{LL}^2 + f_{LS}^2}. \tag{3.2.17}$$

From the expressions of the coupling capacitors, it is obvious that the value of the resistance seen by C_S is much larger than that seen by C_L. Therefore, the frequency f_{LS} is chosen as the dominant frequency with its value equal to f_L. The value of f_{LL} is fixed at one-tenth of f_L in order to reduce its effect on f_L. The values of the capacitors thus obtained are of acceptable sizes, since if f_{LL} were chosen to be equal to f_L, then the value of C_L would have been at least ten times larger than the earlier case. The value of C_S then would have reduced further from its previous value, which is anyway of a much smaller size due to the larger resistance seen by it.

For a given set of performance specifications, the values of the design variables are obtained from our program, and SPICE simulation is carried out in order to verify the design. The results are presented in Table 3.2.2. There are four membership functions used in this case, one each for the four performance specifications, and they are clubbed together in order to formulate the objective function in proportion to their assigned weights. The weight of each membership function is taken to be 0.25, i.e., all the performance specifications are considered to be equally important. The values for the constant parameters V_{CC}, R_S, and R_L are taken to be 12 V, 1 kΩ, and 10 kΩ respectively. Considering f_L to be 100 Hz, the values of C_S and C_L obtained from the design are 0.0385 μF and 0.1587 μF respectively.

Table 3.2.2

The optimization result of the emitter-follower as an output stage along with the results obtained from SPICE simulation.

Sl. No.	*Performance specifications*	*Type*	*Weight*	*Target value*	*Value obtained from the design*	*Value obtained from SPICE simulation*
1.	V_{opp} (V)	≥	0.25	8	8.513	8.6
2.	*delV* (V)	=	0.25	0	0.001	0.05
3.	R_{in} (kΩ)	≥	0.25	400	412.77	411.64
4.	R_{out} (Ω)	≤	0.25	40	31.384	33.78

The results obtained from our optimization routine match closely with those obtained from SPICE simulation. The variations of the performance specifications while arriving at the optimal design point through successive iterations are given in Table 3.2.3. The zeroth order iteration is performed with the initial values of design the variables as supplied by the designer.

Table 3.2.3

Variation of the performance parameters and the overall membership function with successive iterations for the emitter-follower.

Iteration No.	*Design parameters*		*Performance specifications*				*Membership function*
	R_B (kΩ)	R_E (kΩ)	V_{opp} (V)	*delV* (V)	R_{in} (kΩ)	R_{out} (Ω)	μ_D
0.	200	5.0	11.722	2.324	101.432	18.732	0.422
1.	772.84	7.716	8.673	0.002	412.697	31.776	0.924
2.	772.979	7.723	8.513	0.001	412.77	31.384	0.998

In order to verify that our optimizer has produced the global optimum solution, the overall membership function is plotted in three-dimensions with the variation of R_E and R_B from 1 kΩ to 10 kΩ and 100 kΩ to 1 MΩ respectively, as shown in the Fig.3.2.2. It is evident from the figure that even if there are some local maxima exist for the membership function, e.g., A, B, etc., the optimizer reached the global maximum, i.e. point G. The values of the design variables R_E and R_B correspond to the point X, which is found to be having the same values as predicted by the optimizer developed in this work.

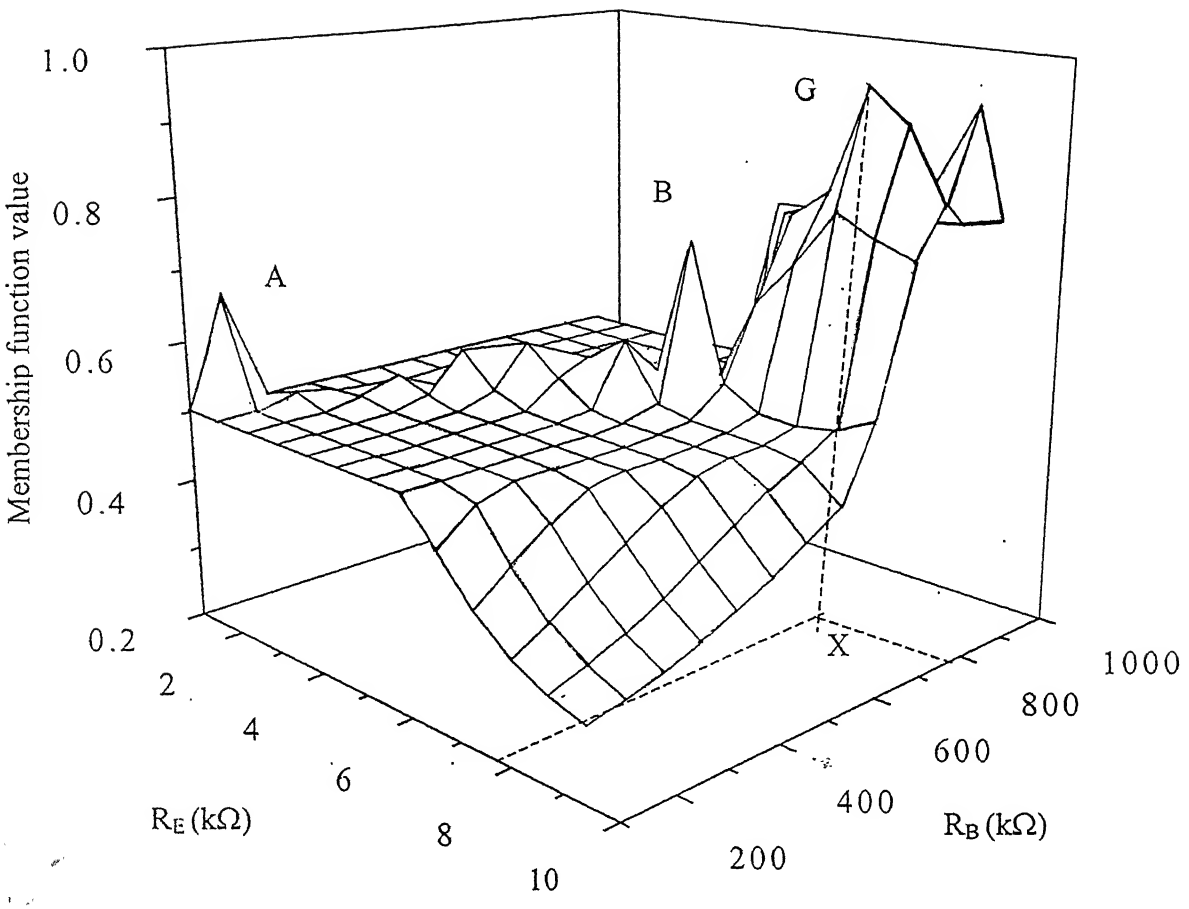


Fig.3.2.2: The variation of the overall membership function for the designed emitter–follower as an output stage in the design space with the variation of R_E from 1 kΩ to 10 kΩ and R_B from 100 kΩ to 1 MΩ.

3.3 THE COMMON-EMITTER STAGE AS AN AMPLIFIER

The schematic of a single-stage common-emitter circuit used as an amplifier is shown in Fig.3.3.1. The performance specifications for which it is optimized are listed in Table 3.3.1.

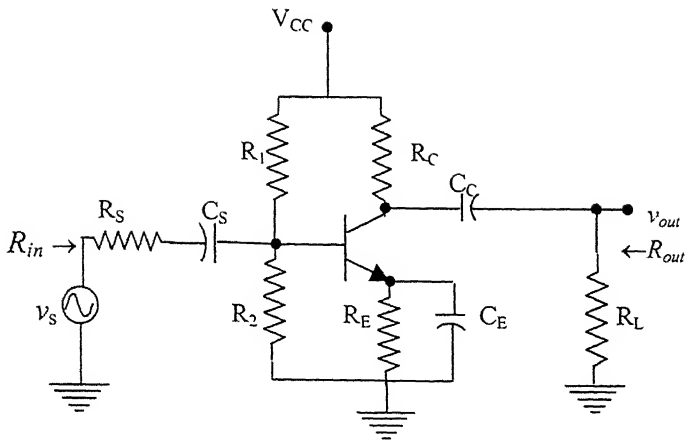


Fig.3.3.1: Schematic of a common-emitter stage used as an amplifier.

Table 3.3.1

The set of performance specifications for which the common-emitter stage as an amplifier is optimized in this work.

Sl. No.	*Symbol*	*Specification*	*Typical objective*
1.	A_v	Voltage gain	Maximize
2.	BW	Bandwidth	Maximize
3.	R_{in}	Input resistance	Maximize/Threshold
4.	R_{out}	Output resistance	Maximize/Threshold

The design variables are the biasing resistors R_1, R_2, R_E, R_C, and the coupling capacitors C_C, C_S, and C_E. However, the biasing resistors are considered to be independent design variables. The values of the coupling capacitors are determined from the consideration of the lower-cutoff frequency. The source resistance R_S, the load resistance R_L, and the supply voltage V_{CC} are taken to be constants by the program. The mathematical expressions used for the computation of the operating point, the small signal parameters, and the performance parameters are given below.

1. The base current (I_B) is given by

$$I_B = \frac{V_{TH} - V_{BE}}{R_{TH} + (\beta_F + 1)R_E}, \tag{3.3.1}$$

where $V_{TH} = V_{CC}R_2/(R_1+R_2)$, and $R_{TH} = R_1 \| R_2$.

2. The transconductance parameter (g_m) is obtained using Eqn.(3.2.3).

3. The input resistance (r_π) of the transistor is determined using Eqn.(3.2.4).

4. The output resistance (r_0) of the transistor is given by

$$r_O = \frac{V_A}{I_C}, \tag{3.3.2}$$

where V_A is the Early voltage (a technology dependant parameter, typically 130 V for *npn* transistors).

5. The input resistance (R_{in}) of the circuit is given by

$$R_{in} = R_S + (r_\pi \| R_{TH}). \tag{3.3.3}$$

6. The output resistance (R_{out}) of the circuit is given by

$$R_{out} = r_O \| R_C \| R_L. \tag{3.3.4}$$

7. The voltage gain (A_V) of the amplifier is given by

$$A_V = \frac{v_{out}}{v_s} = -\frac{R_{TH} \| r_\pi}{R_S + (R_{TH} \| r_\pi)} \frac{\beta_0 R_{out}}{r_\pi}. \tag{3.3.5}$$

8. The bandwidth (BW) of the amplifier is given by

$$BW = f_H - f_L, \tag{3.3.6}$$

1. The base current (I_B) is given by

$$I_B = \frac{V_{TH} - V_{BE}}{R_{TH} + (\beta_F + 1)R_E}, \tag{3.3.1}$$

where $V_{TH} = V_{CC}R_2/(R_1+R_2)$, and $R_{TH} = R_1 \| R_2$.

2. The transconductance parameter (g_m) is obtained using Eqn.(3.2.3).

3. The input resistance (r_π) of the transistor is determined using Eqn.(3.2.4).

4. The output resistance (r_O) of the transistor is given by

$$r_O = \frac{V_A}{I_C}, \tag{3.3.2}$$

where V_A is the Early voltage (a technology dependant parameter, typically 130 V for *npn* transistors).

5. The input resistance (R_{in}) of the circuit is given by

$$R_{in} = R_S + (r_\pi \| R_{TH}). \tag{3.3.3}$$

6. The output resistance (R_{out}) of the circuit is given by

$$R_{out} = r_O \| R_C \| R_L. \tag{3.3.4}$$

7. The voltage gain (A_V) of the amplifier is given by

$$A_V = \frac{v_{out}}{v_s} = -\frac{R_{TH} \| r_\pi}{R_S + (R_{TH} \| r_\pi)} \frac{\beta_0 R_{out}}{r_\pi}. \tag{3.3.5}$$

8. The bandwidth (BW) of the amplifier is given by

$$BW = f_H - f_L, \tag{3.3.6}$$

where f_H is the upper cutoff frequency, given by

$$f_H = \frac{1}{2\pi(R_\mu^0 C_\mu + R_\pi^0 C_\pi)}, \tag{3.3.7}$$

with $R_\mu^0 = R_\pi^0 + (r_0 \,||R_C\,||R_L) + g_m R_\pi^0 (r_0 \,||R_C\,||R_L)$,

and $R_\pi^0 = r_\pi \,||R_S\,||R_{TH}$.

The capacitances are given by [42]

$$C_\mu = \frac{C_{\mu 0}}{\left(1 - \frac{V_{bc}}{V_{jc}}\right)^m}, \tag{3.3.8}$$

where C_μ is the base-collector junction capacitance, $C_{\mu 0}$ is the base-collector junction capacitance at zero bias voltage, V_{bc} is the forward bias voltage on the base-collector junction, V_{jc} is the built-in potential of the base-collector junction (typically 0.5 – 0.7 V), m is the grading coefficient of the junction (typically 0.3–0.5), and

$$C_\pi = C_{je} + C_b, \tag{3.3.9}$$

where C_{je} ($\approx 2C_{je0}$) is the base-emitter junction capacitance [42], with C_{je0} being the base-emitter junction capacitance at zero bias, and C_b ($= \tau_F g_m$) is the base-charging capacitance, with τ_F being the base transit time in the forward direction, typically 10 to 500 ps for *npn* transistors [42].

The lower cutoff frequency (f_L) is generally specified by the user, and its typical value is in the range of 50 to 100 Hz. The values of the bypass capacitors, i.e., C_E, C_S, and C_C are calculated once the values for the biasing resistors are obtained, which satisfy the performance specifications as specified by the user, by using the following algorithm.

1. The expression for C_C is given by

$$C_C = \frac{1}{2\pi R_{CC} f_{LC}}, \tag{3.3.10}$$

where f_{LC} is the lower cutoff frequency due to C_C, and R_{CC} is the resistance seen by C_C, given by

$$R_{CC} = (r_o || R_C) + R_L. \tag{3.3.11}$$

2. The expression for C_S is given by

$$C_S = \frac{1}{2\pi R_{CS} f_{LS}}, \tag{3.3.12}$$

where f_{LS} is the lower cutoff frequency due to C_S, and R_{CS} is the resistance seen by C_S, given by

$$R_{CS} = (r_\pi || R_{TH}) + R_S. \tag{3.3.13}$$

3. The expression for C_E is given by

$$C_E = \frac{1}{2\pi R_{CE} f_{LE}}, \tag{3.3.14}$$

where f_{LE} is the lower cutoff frequency due to C_E, and R_{CE} is the resistance seen by C_E, given by

$$R_{CE} = R_E || \left(\frac{(R_S || R_{TH}) + r_\pi}{\beta_0 + 1} \right). \tag{3.3.15}$$

The lower cutoff frequency (f_L) of the circuit is determined from the consideration of the individual lower cutoff frequencies contributed by each of the three coupling capacitors, and can be given by

$$f_L = \sqrt{f_{LE}^2 + f_{LC}^2 + f_{LS}^2}. \tag{3.3.16}$$

From the expressions of the resistances seen by the capacitors, it is evident that the resistance seen by C_E is much smaller in comparison with the resistances seen by the other two capacitors. In order to get the capacitors of acceptable sizes, the time constant due to C_E is taken to be the dominant one, and f_{LE} is chosen to be equal to f_L. The cutoff frequencies due to other two capacitors, i.e., f_{LC} and f_{LS} are fixed at one-tenth of f_{LE}, in order to reduce their effects on f_L.

For a given set of performance specifications, the values for the design variables of this circuit are obtained and fed to a SPICE file. The values of the supply voltage V_{CC}, the source resistance R_S, and the load resistance R_L are taken to be 12 V, 1 kΩ, and 10 kΩ respectively. The value of f_L is taken to be 100 Hz. The results obtained from our algorithm and SPICE simulation are presented in Table 3.3.2. The values of the design variables obtained from our algorithm, and subsequently used for SPICE simulation are presented in Table 3.3.3. The validity of our design can be authenticated from the fact that the results obtained from the optimization routine match reasonably well with those obtained from SPICE simulation. However, it may be noted that for the computation of bandwidth, simplified equations are used.

Table 3.3.2

The output results obtained from optimization of the common-emitter stage as an amplifier and their comparison with those obtained from SPICE simulation.

Sl. No.	*Performance specification*	*Type*	*Weight*	*Target*	*Value obtained from the design*	*Value obtained from SPICE simulation*
1.	A_V (dB)	≥	0.25	40	39.997	40.53
2.	BW (MHz)	≥	0.25	6.0	6.000	6.12
3.	R_{in} (kΩ)	≥	0.25	6.0	9.488	9.369
4.	R_{out} (kΩ)	≥	0.25	5.0	5.327	5.326

Table 3.3.3

Optimized values of the design variables obtained from our optimization algorithm for the common-emitter amplifier.

Sl. No.	*Parameter*	*Initial guess*	*Final value*
1.	R_E (kΩ)	2.0	3.392
2.	R_C (kΩ)	5.0	11.971
3.	R_1 (kΩ)	100.0	280.011
4.	R_2 (kΩ)	50.0	106.709
5.	C_E (μF)	–	30.72
6.	C_S (μF)	–	1.667
7.	C_C (μF)	–	0.744

3.4 THE COMMON-BASE STAGE AS AN AMPLIFIER

The schematic representation of a common-base stage, which can be used as an amplifier, is shown in Fig.3.4.1. The set of the performance specifications for which it is optimized is the same as those for a common-emitter stage, listed in Table 3.3.1.

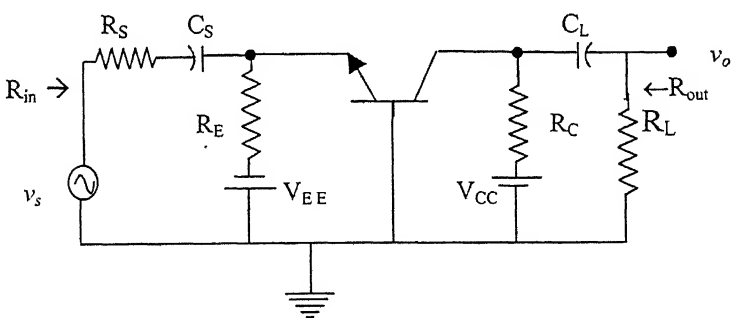


Fig. 3.4.1: Schematic of a common-base stage used as an amplifier.

The design variables are the biasing resistors R_E and R_C, and the coupling capacitors C_S and C_L. The biasing resistors R_E and R_C are considered to be independent design variables. The values of the coupling capacitors are determined from the consideration of

the lower-cutoff frequency. The source resistance R_S, the load resistance R_L, and the supply voltages V_{CC} and V_{EE} are taken to be constants specified by the user. The mathematical expressions used for the computation of the operating point, the small signal parameters, and the performance parameters are given below.

1. The emitter current (I_E) is given by

$$I_E = \frac{V_{EE} - V_{BE}}{R_E}. \tag{3.4.1}$$

2. The input resistance (r_E) of the transistor in this configuration is given by

$$r_E = \frac{V_T}{I_C}. \tag{3.4.2}$$

3. The input resistance (R_{in}) of the circuit is given by

$$R_{in} = R_S + (R_E \| r_E). \tag{3.4.3}$$

4. The output resistance (R_{out}) of the circuit is given by

$$R_{out} = R_C \| R_L. \tag{3.4.4}$$

5. The voltage gain (A_V) of the circuit is given by

$$A_V = \frac{v_o}{v_s} = \frac{R_E \| r_E}{R_S + (R_E \| r_E)} \frac{R_C \| R_L}{r_E}. \tag{3.4.5}$$

6. The bandwidth (BW) of the amplifierm is given by

$$BW = f_H - f_L, \tag{3.4.6}$$

where f_H is the upper cutoff frequency, given by

$$f_H = \frac{1}{2\pi(R_\pi^0 C_\pi + R_\mu^0 C_\mu)},$$

with $R_{\mu}^{0} = (R_C \,||R_L)$, and

$R_{\pi}^{0} = r_E \,||\, R_E||R_S$.

The values of C_S and C_L are determined from the values of R_E and R_C, using the expressions given by Eqns.(3.2.14–3.2.16), substituting the values of R_{CL} and R_{CS} by the following equations

$$R_{CS} = R_S + (R_E \,||\, r_E), \text{ and} \tag{3.4.7}$$

$$R_{CL} = R_C + R_L. \tag{3.4.8}$$

The value of the resistance seen by C_L (i.e., R_{CL}) is much larger than that seen by C_S (i.e., R_{CS}). Therefore, f_{LS} is considered to be the dominant frequency and its value is chosen to be equal to f_L.

For a given set of performance specifications, the values for the design variables of the common-base amplifier stage are obtained and fed to a SPICE file. The results obtained from our program and SPICE simulation are presented in Table 3.4.1. The values of the constants V_{CC}, V_{EE}, R_S, and R_L used in the optimization program are 10 V, –2 V, 50 Ω, and 10 kΩ respectively. The value of f_L is taken to be 100 Hz. The values of R_S and R_L are chosen considering the fact that the gain of the common-base amplifier depends to a large extent on these parameters. The design variables used as the initial guess and those obtained from the optimization algorithm are listed in Table 3.4.2.

The results obtained from our optimization algorithm match well with those obtained from SPICE simulation, except for the bandwidth of the amplifier. This is due to the pole splitting caused by the fact that the two poles created due to C_μ and C_π are very close to each other. Thus, the value of the bandwidth obtained from SPICE simulation is higher than that obtained from our optimization algorithm.

Table 3.4.1

The output results obtained from the optimization of the common-base stage as an amplifier for a given set of specifications and those obtained from the SPICE simulation.

Sl. No.	*Performance specification*	*Type*	*Weight*	*Target value*	*Value obtained from the design*	*Value obtained from SPICE simulation*
1.	A_V (dB)	≥	0.25	30	30.268	30.40
2.	BW (MHz)	≥	0.25	100	317.617	398.10
3.	R_{in} (Ω)	≥	0.25	100	123.03	120.48
4.	R_{out} (kΩ)	≥	0.25	4	4.092	4.066

Table 3.4.2

The design variables obtained from the optimization of the common-base stage as an amplifier along with the initial values used for the algorithm.

Sl. No.	*Parameter*	*Initial guess*	*Final value*
1.	R_E (kΩ)	3	3.723
2.	R_C (kΩ)	5	6.926
3.	C_S (μF)	–	12.94
4.	C_L (μF)	–	0.94

CHAPTER 4

OPTIMIZATION OF SIMPLE MOS CIRCUITS

4.1. INTRODUCTION

In order to examine the validity of the fuzzy optimization approach, as discussed in Chapter 2, an attempt has been made to apply it for the synthesis of simple analog MOS circuits. This chapter deals with this aspect of the work for two simple analog MOS building blocks, e.g., current mirrors and common-source amplifiers, which are used as modules for op-amps. The technology parameters, e.g., the nominal threshold voltage of the transistor (V_{to}), oxide capacitance per unit area (C'_{ox}), minimum value of channel length (L_{drawn}), etc., are taken from a 0.8 μ *n*-well process [42], the summary of which is given in Appendix–III. The specifications and the performance objectives, for which the above-mentioned elementary building blocks are optimized, are described in this chapter. The mathematical equations used for the computation, and the design results obtained for a set of performance specifications for each of the topologies are also presented.

4.2 CURRENT MIRRORS

The current mirror, also referred to as current source/sink, is the most basic building block in CMOS IC design and is used in various analog circuits. Ideally, the output impedance of a current source/sink should be infinite, and it should be capable of supplying/drawing a constant current over a wide range of voltages across it. However, in reality, the performance of all practical current mirrors is limited by finite values of the output impedance, and smaller output voltage swing (in order to keep the devices in

saturation). Depending on the requirements, a particular current mirror is chosen keeping in mind the performance it can provide.

The following different types of current mirrors are designed and simulated in this work.

1. Simple current mirror.
2. Cascode current source.
3. Wilson current source.
4. Modified Wilson current source.
5. Regulated cascode current source.

For the design of current mirrors, the performance objectives are to minimize the total area, a large output resistance for a given current, and a more or less constant current for an output voltage range specified by the designer. This is presented in Table 4.2.1.

Table 4.2.1

The performance specifications and objectives for synthesis of current mirrors.

Sl. No.	*Symbol*	*Specification*	*Typical objective*
1.	I_{out}	Output current	Threshold
2.	V_{outmin}	Minimum output voltage	Minimize
3.	R_{out}	Output resistance	Maximize
4.	*AREA*	Active area	Minimize

The design variables are the width (W) and the length (L) of the transistors, and the bias current (I_{bias}). The bias voltage (V_{bias}) is a design variable for the regulated cascode current source only. The supply voltages V_{DD} and V_{SS} are constants supplied by the user, along with the maximum possible length (L_{max}) and width (W_{max}) of the transistors. For all the design of current sources presented in this work, supply voltages of $\pm$ 3 V are used. In order to reduce the total number of independent variables for the optimization routine, all the transistors in a particular current source are chosen to have the same size, i.e., they have the same lengths and the same widths. Thus, the bias current I_{bias} is equal to the output

current I_{out} for all cases considered. The following approaches could be implemented to replace the bias current source (I_{bias}).

1. A resistor.
2. A depletion-load NMOS.
3. A pseudo-NMOS load.

We have simulated the bias current (I_{bias}) using a pseudo NMOS-load (i.e., a PMOS with its gate grounded), as shown in Fig.4.2.1 for all the current sources. Thus, the length (L_0) and the width (W_0) of the load transistor are also considered to be variables for the optimization algorithm.

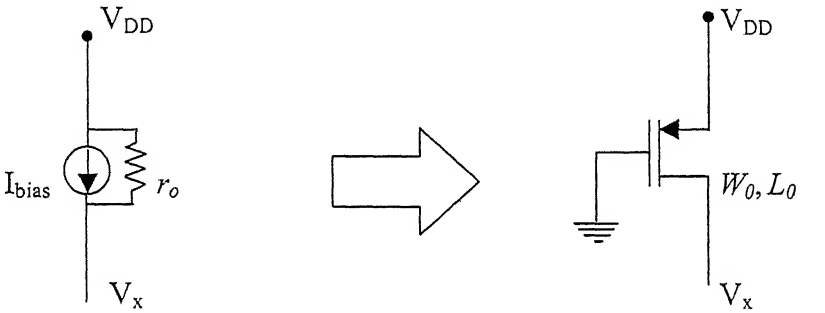


Fig.4.2.1: A representation of the bias current source by a pseudo-NMOS load, as implemented in our design.

The values for the bias current I_{bias} and the voltage V_x are obtained from the optimization routine for the synthesis of current mirrors. The (W/L) ratio is obtained from the current expression, using the known values for the gate-source voltage (V_{GS}), the drain-source voltage (V_{DS}), and the drain current (I_D). Assigning L_0 or W_0 a value equal to that for the minimum channel length (L_{drawn}), according to the W_0/L_0 ratio being greater than or less than unity respectively, the other one is determined.

We have used the Shichman-Hodges model [41] for all computational purposes. Once the operating point is determined, the small signal parameters are calculated using the following set of equations [43].

1. The transconductance (g_m) is given by

$$g_m = \sqrt{2k'\left(\frac{W}{L}\right)I_D(1+\lambda V_{DS})}, \tag{4.2.1}$$

where $k'(=\mu C'_{ox})$ is the process transconductance parameter, μ is the carrier mobility, C'_{ox} is the gate-oxide capacitance per unit area, W/L is the aspect ratio, λ is the channel length modulation parameter, which depends on the effective channel length L_{eff} $(=L - X_d)$ of the transistor, and is given by [42]

$$\lambda = \frac{1}{L_{eff}}\frac{dX_d}{dV_{DS}}, \tag{4.2.2}$$

where X_d is the drain depletion width, and dX_d/dV_{DS} is the rate of variation of X_d with V_{DS} (a technology dependent parameter).

2. The drain conductance (g_d) is given by

$$g_d = \frac{\lambda I_D}{(1+\lambda V_{DS})}. \tag{4.2.3}$$

3. The body transconductance (g_{mb}) is given by

$$g_{mb} = \chi g_m, \tag{4.2.4}$$

where χ is the body transconductance factor given by [43]

$$\chi = -\frac{\gamma}{2\sqrt{2\varphi_F + V_{SB}}}, \tag{4.2.5}$$

where φ_F $(= V_T ln \frac{N_A}{n_i})$ is the bulk potential of the substrate, with V_T being the thermal voltage, N_A being the substrate doping, and n_i being the intrinsic carrier concentration; γ $(=\sqrt{2q\varepsilon_S N_A}/C'_{ox})$ is the body effect coefficient, with ε_S being the permittivity of Si, and V_{SB} is the source-body potential.

4.2.1 SIMPLE CURRENT MIRROR

The schematic of a simple current mirror, optimized for a given set of performance specifications, is shown in Fig.4.2.1.1 along with the node numbers, which are used to generate the SPICE netlist.

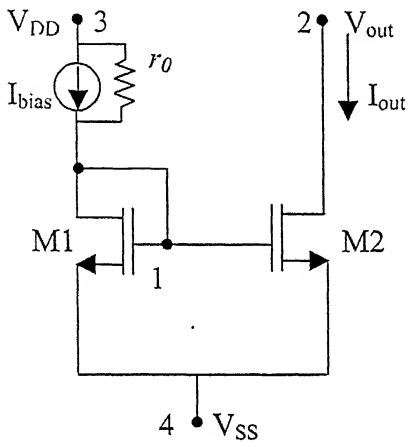


Fig. 4.2.1.1: Schematic of a simple current mirror.

The mathematical equations and the concepts used for the computation of the performance specifications are given below.

1. The output resistance (R_{out}) is given by

$$R_{out} = r_{o2} = \frac{1+\lambda V_{out}}{\lambda I_{out}}. \qquad (4.2.1.1)$$

2. The minimum output voltage (V_{outmin}) is given by

$$V_{outmin} = V_{GS} - V_{tno}, \qquad (4.2.1.2)$$

where V_{GS} is the gate-to-source voltage of the transistors M1 and M2, and V_{tno} is the threshold voltage of the transistors at zero back bias.

3. The active area taken up by the current mirror is given by

$$AREA = 2\,W \times L + W_0 \times L_0. \qquad (4.2.1.3)$$

Based on the above mentioned assumptions and equations, a module has been developed for the simple current mirror, which gives the design variables as output from a given set of performance specifications as input to the routine. The output results along with the results obtained from SPICE simulation, and the design parameters obtained for a given set of performance specifications are shown in Tables 4.2.1.1 and 4.2.1.2 respectively. The results obtained from our algorithm match well with those obtained from SPICE simulation.

Table 4.2.1.1

The design results obtained for a simple current mirror and their comparison with those obtained from SPICE simulation.

Sl. No.	*Performance specification*	*Type*	*Weight*	*Target*	*Value obtained from design*	*Value obtained from SPICE simulation*
1.	I_{out} (μA)	=	0.25	20	20	20.12
2.	R_{out} (MΩ)	≥	0.25	1	1.645	1.658
3.	V_{outmin} (Volts)	≤	0.25	0.3	0.207	0.21
4.	*AREA* (μ^2)	≤	0.25	100	98.56	–

Table 4.2.1.2.

The design variables obtained from our optimization program for a simple current mirror.

Sl. No.	*Design variable*	*Symbol*	*Value*
1.	Bias current (μA)	I_{bias}	20
2.	Width/length of M1, M2 (μm/μm)	*W/L*	18.293/2.559
3.	Width/length of load transistor (μm/μm)	W_0/L_0	0.8/6.286

4.2.2 CASCODE CURRENT SOURCE

The cascode current source is a combination of two simple current mirrors, as shown in Fig.4.2.2.1. The node numbers used to generate the SPICE netlist are also shown in the figure. This configuration leads to a tremendous increase in the output resistance, at the cost of a reduced output voltage swing. The bodies of all the transistors are connected to V_{SS}, and, therefore, the body effect of the two transistors M3 and M4 is considered in the analysis.

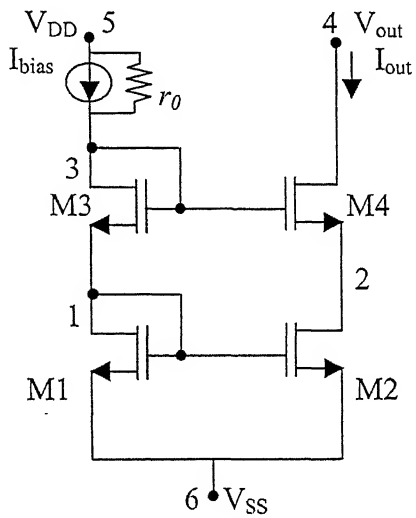


Fig. 4.2.2.1: Schematic of a cascode current source.

The mathematical equations and the concepts used to compute the performance specifications are given below.

1. The output resistance (R_{out}) of the current source is given by [41]

$$R_{out} = r_{o4}\left(1+\left(g_{m4}+g_{mb4}\right)r_{o2}\right)+r_{o2}. \tag{4.2.2.1}$$

2. The minimum output voltage (V_{outmin}) is given by

$$V_{outmin} = V_{GS2} + V_{GS4} - V_{tn4}, \tag{4.2.2.2}$$

where V_{GS2} and V_{GS4} are the gate-source voltages of M2 and M4 respectively, and V_{tn4} is the threshold voltage of M4 with body effect.

3. The active area taken up by the current source is given by

$$AREA = 4\,W \times L + W_0 \times L_0. \tag{4.2.2.3}$$

Based on the above assumptions and equations, a module has been developed, which gives the design variables as output from a given set of performance specifications as input to the routine. The output results along with the results obtained from SPICE simulation, and the design variables obtained for a given set of performance specifications are shown in Tables 4.2.2.1 and 4.2.2.2 respectively. The authenticity of our design is evident from the fact that the results obtained from our design closely match with those obtained from SPICE simulation.

Table 4.2.2.1

The design results obtained for a cascode current source and their comparison with the results obtained from SPICE simulation.

Sl. No.	*Performance specification*	*Type*	*Weight*	*Target*	*Value obtained from design*	*Value obtained from SPICE Simulation*
1.	I_{out} (μA)	=	0.25	20	20	20.026
2.	R_{out} (MΩ)	≥	0.25	100	120.8	185.29
3.	V_{outmin} (Volts)	≤	0.25	1.2	0.981	1.0
4.	$AREA$ (μ^2)	≤	0.25	100	57.271	–

Table 4.2.2.2

The design variables obtained from our optimization routine for a cascode current source.

Sl. No.	*Design variable*	*Symbol*	*Value*
1.	Bias current (μA)	I_{bias}	20.0
2.	Width/length of M1, M2, M3, and M4 (μm/μm)	W/L	13.971/0.935
3.	Width/length of load transistor (μm/μm)	W_0/L_0	0.8/6.225

4.2.3 WILSON CURRENT SOURCE

The basic current mirror is improved significantly with negative feedback. The Wilson current source uses this concept to offer stable current values for wide voltage swings and enhanced output resistance. Figure 4.2.3.1 shows the schematic of a Wilson current source considered in this work along with the node numbers which are used to generate the SPICE netlist. All the bodies of the transistors are connected to V_{SS}.

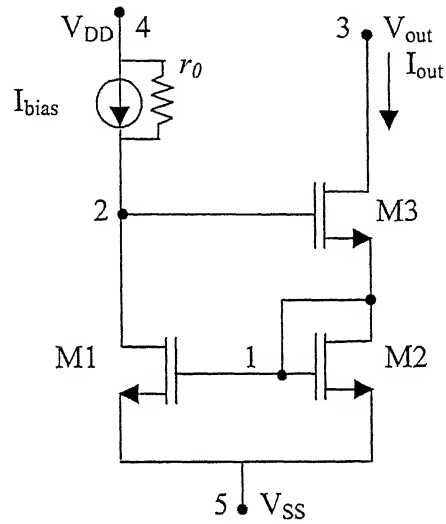


Fig. 4.2.3.1: Schematic of a Wilson current source.

The performances of this current source are computed using the following mathematical expressions.

1. The output resistance (R_{out}) is given by [41]

$$R_{out} = r_{o3}\left[1 + g_{m1}(r_{o1} \,||\, r_0) + g_{mb3}\left(\frac{1}{g_{m2}}\right) + \frac{1}{r_{o3}g_{m2}}\right]. \tag{4.2.3.1}$$

2. The minimum output voltage (V_{outmin}) is given by

$$V_{outmin} = V_{GS2} + V_{DS3,sat} = V_{GS2} + V_{GS3} - V_{tn3}. \tag{4.2.3.2}$$

3. The active area taken up by the current mirror can be given by

$$AREA = 3\,W \times L + W_0 \times L_0. \tag{4.2.3.3}$$

Based on the above assumptions and equations, a module has been developed that gives the design variables as output from a given set of performance specifications as input to the routine. The output results along with the results obtained from SPICE simulation for a given set of performance specifications are shown in Table 4.2.3.1. The design variables obtained are given in Table 4.2.3.2.

Table 4.2.3.1

The design results for a Wilson current source and their comparison with the results obtained from SPICE simulation.

Sl. No.	*Performance specification*	*Type*	*Weight*	*Target*	*Value obtained from design*	*Value obtained from SPICE Simulation*
1.	I_{out} (μA)	=	0.25	20	20	18.89
2.	R_{out} (MΩ)	≥	0.25	100	132.9	198.42
3.	V_{outmin} (Volts)	≤	0.25	1.2	1.086	1.07
4.	*AREA* (μ^2)	≤	0.25	100	40.309	–

The results obtained from our design using the tool developed in this work match reasonably well with those obtained from SPICE simulation. However, it is to be noted that the current through the output branch is not equal to that of the input branch due to the difference in the drain-source voltages across the respective transistors and the channel length modulation effect.

Table 4.2.3.2

The design variables obtained from our optimization program for a Wilson current source.

Sl. No.	*Design Variable*	*Symbol*	*Value*
1.	Bias current (μA)	I_{bias}	20
2.	Width/length of M1, M2, and M3 (μm/μm)	*W/L*	9.711/1.212
3.	Width/length of load transistor (μm/μm)	W_0/L_0	0.8/6.251

4.2.4 MODIFIED WILSON CURRENT SOURCE

The schematic of the modified Wilson current source along with the node numbers used to generate the SPICE netlist are shown in Fig.4.2.4.1. For the case of the Wilson current source, the current through the input and the output branches are not same due to the mismatch in the drain-source voltages, as mentioned earlier. This configuration eliminates that problem by putting a diode-connected transistor M4 in the input branch as well, as shown in the figure, and, thus, the voltages at nodes 3 and 4 are always equal to each other.

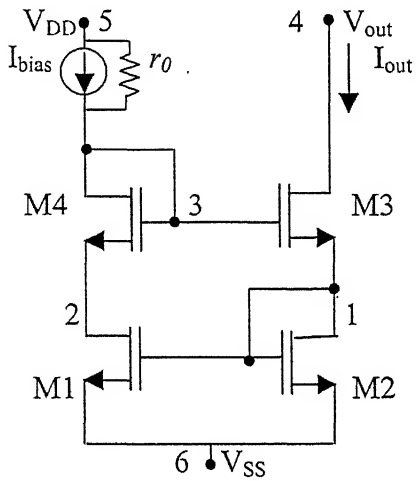


Fig. 4.2.4.1: Schematic of the modified Wilson current source.

The mathematical equations and the concepts used to compute the performance specifications are given below.

1. The output resistance (R_{out}) and the minimum output voltage (V_{outmin}) of the modified Wilson current source are the same as those for the Wilson current source and can be calculated using Eqns.(4.2.3.1) and (4.2.3.2) respectively.

2. The active area taken up by the current mirror is given by

$$AREA = 4\,W \times L + W_0 \times L_0. \tag{4.2.4.1}$$

Based on the above assumptions and equations, a module has been developed, which gives the design variables as output from a given set of performance specifications as input to the routine. The output results along with the results obtained from SPICE simulation for a given set of performance specifications are shown in Table 4.2.4.1. The design variables obtained are given in Table 4.2.4.2. The results obtained from our design match reasonably well with those obtained from SPICE simulation.

Table 4.2.4.1

The design results obtained for the modified Wilson current source and their comparison with the results obtained from SPICE simulation.

Sl. No.	*Performance specification*	*Type*	*Weight*	*Target*	*Value obtained from design*	*Value obtained from SPICE Simulation*
1.	I_{out} (μA)	=	0.25	20	20	20.06
2.	R_{out} (MΩ)	≥	0.25	100	109.8	142.05
3.	V_{outmin} (V)	≤	0.25	1.2	1.036	1.03
4.	*AREA* (μ^2)	≤	0.25	100	32.243	–

Table 4.2.4.2

The design variables obtained from our optimization program for the modified Wilson current source.

Sl. No.	*Design variable*	*Symbol*	*Value*
1.	Bias current (μA)	I_{bias}	20
2.	Width/length of M1, M2, M3, and M4 (μm/μm)	*W/L*	9.71/0.935
3.	Width/length of load transistor (μm/μm)	W_0/L_0	0.8/6.253

4.2.5. REGULATED CASCODE CURRENT SOURCE

The regulated cascode current source uses negative feedback in order to stabilize the output current and increase the output impedance to an even higher degree than the Wilson current source. Figure 4.2.5.1 shows the schematic of a regulated cascode current source along with the node numbers to generate the SPICE netlist.

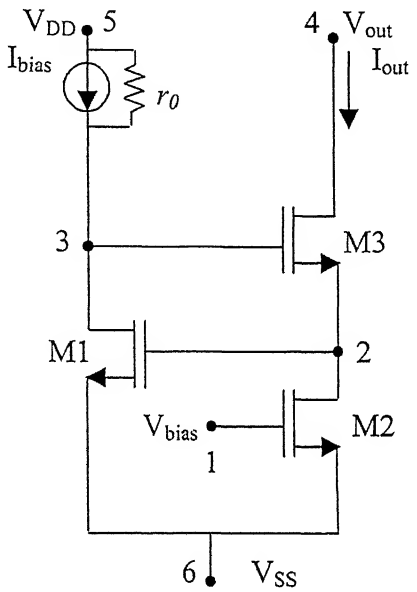


Fig.4.2.5.1: Schematic of a regulated cascode current source.

The transistors M1 and M3 form the negative feedback loop, which stabilizes the output current I_{out}. In order to increase I_{out}, the voltage at node 2 must increase, which in turn increases the current through M1. Since the current through the bias current source I_{bias} is a constant, hence, the voltage at node 3 decreases, thereby offsetting the increase in I_{out} by decreasing the gate-source voltage of M3. The higher value of the output impedance is obtained from the fact that M2 is no longer diode connected as was the case for the Wilson current source. Hence, the output impedance (r_{o2}) of M2 contributes to the output resistance in place of the inverse transconductance parameter ($1/g_{m2}$) for the Wilson current source, thus achieving a much higher multiplication factor.

The mathematical equations and the concepts used to compute the performance specifications are given below.

1. The bias voltage V_{bias} is not an independent design variable, since its value is obtained from the current through M2 and its W/L ratio.

2. The output resistance (R_{out}) is given by [41]

$$R_{out} = r_{o3}\left[1 + g_{m3}r_{o2}\left(1 + g_{m1}\left(r_{o1} \parallel r_0\right)\right) + g_{mb3}\, r_{o2} + \frac{r_{o2}}{r_{o3}}\right]. \qquad (4.2.5.1)$$

3. The minimum output voltage (V_{outmin}) is given by

$$V_{outmin} = V_{GS1} + V_{DS3,sat} = V_{GS1} + V_{GS3} - V_{tn3}. \qquad (4.2.5.2)$$

4. The active area taken up by the current mirror is given by

$$AREA = 3\, W \times L + W_0 \times L_0. \qquad (4.2.5.3)$$

Based on the above assumptions and equations, a module has been developed, which gives the design variables as output from a given set of performance specifications as input to the routine. The output results along with the results obtained from SPICE simulation, and the design variables obtained are given in Tables 4.2.5.1 and 4.2.5.2 respectively.

Table 4.2.5.1

The design results obtained for the regulated cascode current source and their comparison with the results obtained SPICE simulation.

Sl. No.	*Performance specification*	*Type*	*Weight*	*Target*	*Value obtained from design*	*Value obtained from SPICE Simulation*
1.	I_{out} (μA)	=	0.25	20	20	20
2.	R_{out} (GΩ)	≥	0.25	1	13.3	11.6
3.	V_{outmin} (Volts)	≤	0.25	1.2	1.09	1.05
4.	$AREA$ (μ^2)	≤	0.25	100	85.158	–

Table 4.2.5.2

The design variables obtained from our optimization program for the regulated cascode current source.

Sl. No.	*Design variable*	*Symbol*	*Value*
1.	Bias current (μA)	I_{bias}	20
2.	Width/length of M1, M2, and M3 (μm/μm)	W/L	5.279/5.064
3.	Width/length of load transistor (μm/μm)	W_0/L_0	0.8/6.223
4.	Bias voltage of M2 (V)	V_{bias}	-1.755

The results obtained from our design match closely with those obtained from SPICE simulation. A significantly higher value of output impedance is achievable using this type of current source. Another advantage of this configuration is that the minimum value of the output voltage is lower than other configurations, except for that of the simple current mirror.

4.3 THE COMMON-SOURCE AMPLIFIER AS A GAIN STAGE

Single-stage amplifiers are used in virtually every op-amp design. By using an active load, a significant amount of chip area can be saved. At the same time, since an active load presents a higher value of the output resistance, the net result is a much higher gain. The schematic of a CMOS common-source amplifier (NMOS driver with PMOS load) used as a gain stage in shown in Fig.4.3.1. This is an important block in CMOS technology, and is generally used as the second stage in a multi-stage op-amp in order to increase the overall gain. A list of the design objectives and the constraints for which the above circuit topology is optimized are given in Table 4.3.1. The design variables are the widths (W) and the lengths (L) of the transistors M1 and M2, and the dc bias voltages V_{bias1} and V_{bias2}.

The following assumptions are made in order to get the independent variables, which are varied during the optimization process.

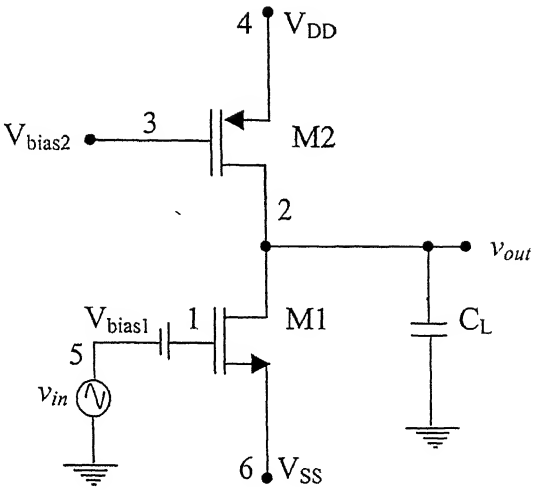


Fig. 4.3.1: Schematic of a CMOS common-source amplifier used as a gain stage.

Table 4.3.1

The list of performance specifications for a CMOS common-source amplifier, used as a gain stage.

Sl. No.	*Symbol*	*Specification*	*Typical objective*
1.	*Gain*	Voltage gain	Maximize
2.	*UGF*	Unity-gain frequency	Maximize
3.	*SR*	Slew rate	Maximize
4.	V_{omax}	Maximum output voltage	Maximize
5.	V_{omin}	Minimum output voltage	Minimize
6.	*PD*	Power dissipation	Minimize
7.	*AREA*	Area	Minimize

1. Out of the six design variables (i.e., W1, L1, W2, L2, V_{GS1}, and V_{GS2}), W1, L1, L2, and V_{bias1} are considered as independent variables.

2. The DC output voltage is taken to be equal to (V_{DD}+V_{SS})/2 (typically zero, when V_{DD} = –V_{SS}), and the bias voltage V_{bias2} is taken to be equal and opposite in sign to V_{bias1} in order to get symmetrical output voltage swing without any clipping.

3. The gate-source voltage of M1 determines the current through the amplifier, and, therefore, the value of the width (W2) of M2 is obtained from the current equation by substituting the known values of V_{bias2}, V_{DS2}, and L2.

The supply voltages V_{DD} and V_{SS} are taken as constant inputs supplied by the user (typical values are 3 V and –3 V respectively). The value of the load capacitance (C_L) is read by the program as a constant input. The mathematical expressions used for the computation of the DC operating point and the performance specifications are given below.

a) *Operating point determination:*

1. The drain-to-source voltages of M1 and M2 are (V_{out} – V_{SS}) and (V_{out} – V_{DD}) respectively.

2. The current I_D through M1 and M2 is determined by the following expression

$$I_D = k_1' \left(\frac{W_1}{L_1} \right) (V_{GS1} - V_{mo})^2 (1 + \lambda V_{DS1}). \tag{4.3.1}$$

After the DC operating point is determined, the small signal parameters are computed using Eqns.(4.2.1– 4.2.4).

b) *Performance objectives computation:*

1. The voltage gain of the amplifier (v_{out}/ v_{in}) is given by

$$Gain = - g_{m1} (r_{o1} \parallel r_{o2}), \tag{4.3.2}$$

where g_{m1} is the transconductance of M1, and r_{o1} and r_{o2} are the drain-source resistances of M1 and M2 respectively.

2. The unity-gain frequency is given by [41]

$$UGF = \frac{g_{m1}}{2\pi C_{LOAD}}, \tag{4.3.3}$$

where $C_{LOAD} = C_L + C_{DB1} + C_{DB2} + C_{GD1} + C_{GD2}$, where C_{DB1} and C_{DB2} are the drain-body capacitances of M1 and M2 respectively, and C_{GD1} and C_{GD2} are the gate-drain capacitances of M1 and M2 respectively.

3. The slew rate of the amplifier is given by [41]

$$SR = I_D (C_{LOAD})^{-1}. \tag{4.3.4}$$

4. The maximum output voltage ($V_{o\max}$) is given by

$$V_{o\max} = V_{DD} - |V_{DSAT2}|. \tag{4.3.5}$$

5. The minimum output voltage ($V_{o\min}$) is given by

$$V_{o\min} = V_{SS} + V_{DSAT1}. \tag{4.3.6}$$

6. The power dissipation (PD) is given by

$$PD = (V_{DD} - V_{SS}) I_D. \tag{4.3.7}$$

7. The active area ($AREA$) is given by

$$AREA = W_1 \times L_1 + W_2 \times L_2. \tag{4.3.8}$$

The common-source amplifier, as shown in Fig.4.3.1, is designed for a set of performance specifications and constraints. The supply voltages V_{DD} and V_{SS} and the load capacitance (C_L) are taken to be ± 3 V and 5 pF respectively. The results obtained are compared with those obtained from SPICE simulation and are shown in Table 4.3.2. The obtained designed variables are given in Table 4.3.3. The variation of the voltage gain with the frequency, as obtained from SPICE simulation for the designed topology is plotted in Fig.4.3.2. The gain and the unity-gain frequency of the amplifier are obtained from the plot

and are given in Table 4.3.2. The design results predicted by our algorithm match reasonably well with those obtained from SPICE simulation.

Table 4.3.2

The output results obtained from the design of a common-source amplifier for a given set of performance specifications along with the results obtained from SPICE simulation.

Sl. No.	*Symbol*	*Type*	*Weight*	*Target*	*Value obtained from design*	*Value obtained from SPICE simulation*
1.	*Gain (dB)*	≥	0.2	30	34.922	34.94
2.	*UGF (MHz)*	≥	0.2	10	10.905	11.2
3.	*SR (V/μs)*	≥	0.2	20	23.356	42.68
4.	V_{omax} *(V)*	≥	0.1	2.3	2.318	2.318
5.	V_{omin} *(V)*	≤	0.1	-2.3	-2.318	2.318
6.	*PD (mW)*	≤	0.1	1	0.759	0.7064
7.	*AREA* (μ^2)	≤	0.1	100	45.448	–

Table 4.3.3

The design variables obtained for a particular set of performance specifications for a common-source amplifier.

Sl. No.	*Design variable*	*Designed value*
1.	W/L of M1 (μm/μm)	10.576/2.866
2.	W/L of M2 (μm/μm)	11.067/1.368
3.	V_{bias1} (V)	-1.618
4.	V_{bias2} (V)	1.618

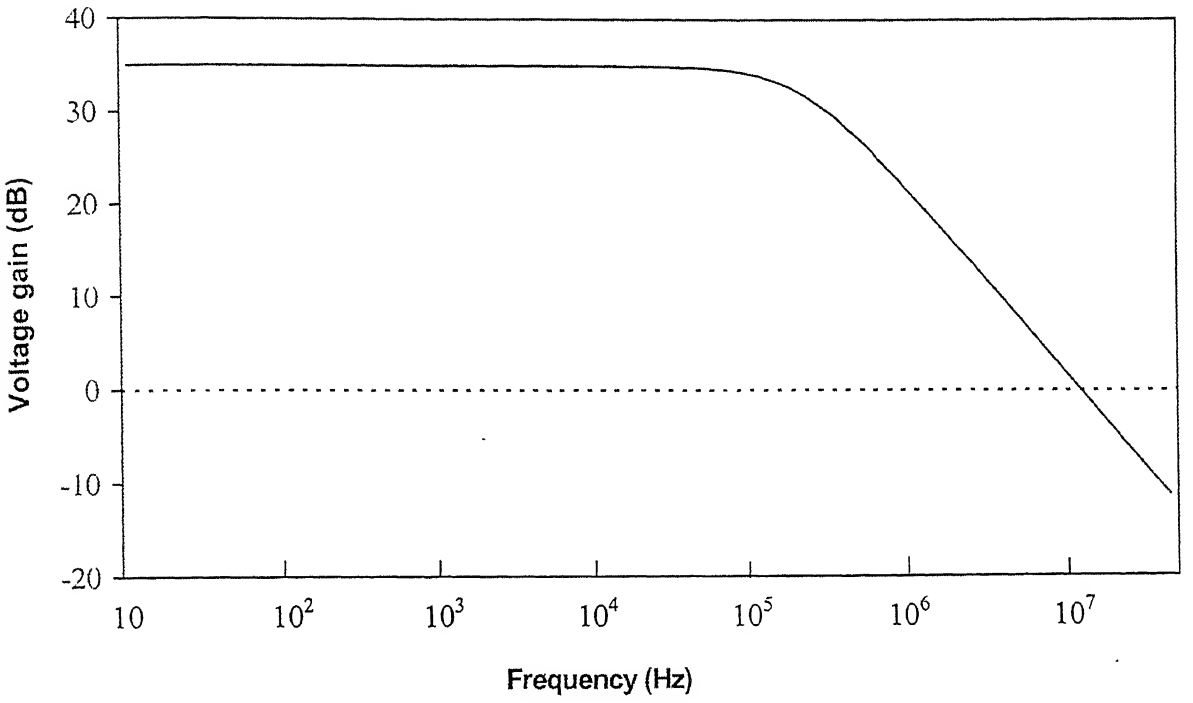


Fig. 4.3.2: The gain versus frequency plot for the common-source amplifier, as obtained from SPICE simulation.

The synthesis procedure for basic MOS analog building blocks, e.g. current sources (simple, cascode, Wilson, modified Wilson, and regulated cascode) and the common–source amplifier as a gain stage are developed and automated in this work. The design results obtained from the optimization routine match well with those obtained from SPICE simulation. A wide variety of simple MOS circuits can be found in [44] for which design routines can be developed using the approach adopted in this work.

CHAPTER 5

OPTIMIZATION OF OPERATIONAL AMPLIFIERS

5.1 INTRODUCTION

The operational amplifier (op-amp) is one of the basic building blocks used for a wide variety of analog circuits, e.g., A/D converters, switched-capacitor filters, etc. Therefore, optimization of the design of the op-amp is of tremendous importance in order to obtain optimized analog circuits, which have more complexity and are in the higher order of hierarchy. In this chapter, the optimization procedures of three basic types of op-amps have been discussed. These are the simple operational transconductance amplifier (OTA), the basic two-stage (BTS) op-amp, and the symmetrical OTA. The simple OTA and BTS op-amps are also known as general-purpose op-amps [45]. The synthesis routines for each of the above mentioned op-amp topologies are described in the respective subsections. The parameters of the circuit (i.e., the design variables), the output results obtained from the optimization routine for a given set of performance specifications, and their comparison with SPICE simulation are also presented.

The set of performance specifications and constraints for which the automation program has been developed in order to optimize all the three basic types of op-amp topologies are listed in Table 5.1.1. However, depending on the objective (i.e., either to minimize/maximize, or to achieve a specific value) as specified by the designer, the membership functions for the performance specifications are suitably framed from the set of functions available in the library.

Table 5.1.1

The set of performance specifications for which the three basic types of op-amps are optimized in our program.

Sl. No.	*Symbol*	*Performance specification*	*Typical objective*
1.	*Gain*	Voltage gain of the amplifier	Maximize
2.	*UGF*	Unity-gain frequency	Maximize
3.	*PM*	Phase margin	Threshold
4.	*SR*	Slew rate	Maximize/Threshold
5.	*CMR+*	Positive common-mode range	Maximize/Threshold
6.	*CMR−*	Negative common-mode range	Minimize/Threshold
7.	V_{outmax}	Maximum output voltage	Maximize/Threshold
8.	V_{outmin}	Minimum output voltage	Minimize/Threshold
9.	*CMRR*	Common-mode rejection ratio	Maximize
10.	$PSRR_{DD}$	Power supply rejection ratio at V_{DD}	Maximize
11.	$PSRR_{SS}$	Power supply rejection ratio at V_{SS}	Maximize
12.	*RMS noise*	Flicker noise as referred to the input	Minimize
13.	*PD*	Power dissipation	Minimize
14.	*AREA*	Active area	Minimize

The computation of the DC operating point is not a straightforward task for MOS circuits when the channel length modulation parameter (λ) is taken into consideration. To obtain the quiescent point, the equations describing the circuit behavior are to be solved iteratively. It is observed that in the CMOS analog circuits, the DC currents through all the transistors are essentially determined by the current through only a few transistors referred to as the current source transistors [41]. Then, from the respective drain currents and the sizes of the transistors, their gate-to-source voltages can be obtained, and, subsequently, the various node voltages can be found. This is performed in an iterative manner, and is represented by the means of a flow chart, as shown in Fig.5.1.1.

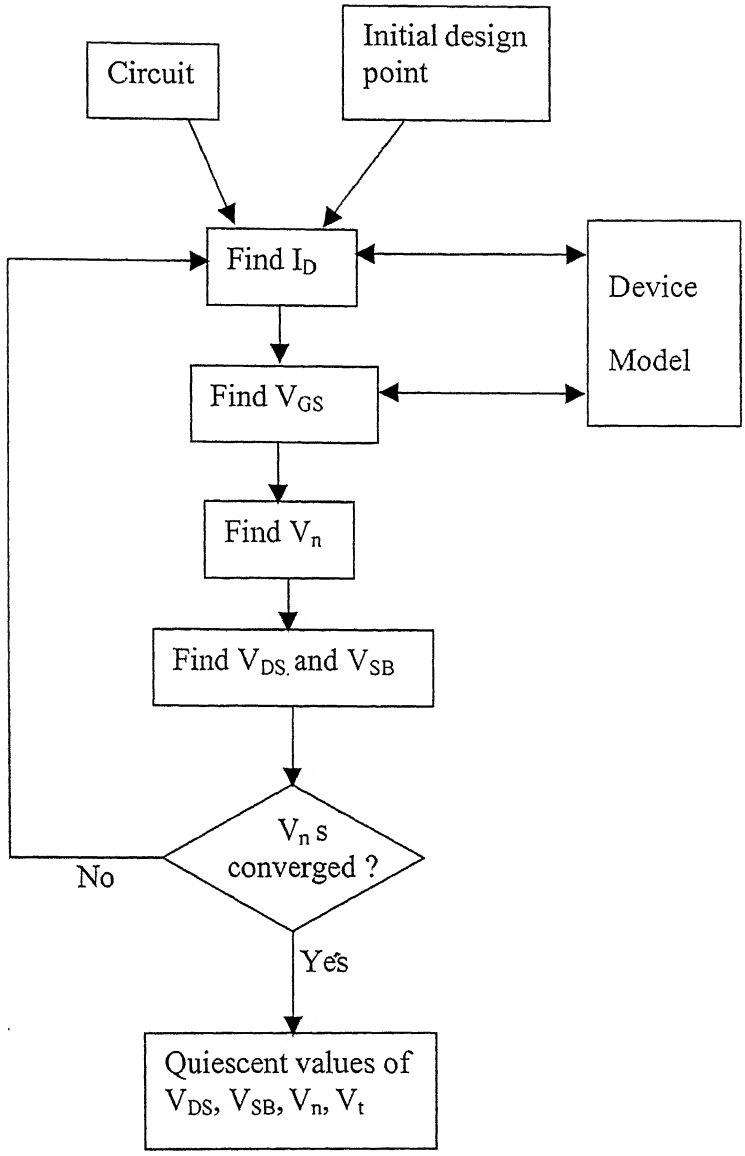


Fig.5.1.1: The flowchart representation of the steps adopted in this work for obtaining the DC operating point.

In the first step, the parameters ($1+\lambda V_{DS}$) and threshold voltages (V_t) are updated based on the V_{DS} and V_{SB} values obtained in the previous iteration using the following expression

$$V_t = V_{to} + \gamma\left(\sqrt{V_{SB} + 2\varphi_F} - \sqrt{2\varphi_F}\right), \tag{5.1}$$

where V_{to} is the threshold voltage of the transistor at zero back bias, and V_t is the threshold voltage with back bias. The parameters V_{to}, γ, and φ_F are technology dependent, and are taken from a technology file (0.8 μ n-well process [42], the summary of which is given in Appendix-III). In the next step, the drain currents (I_D) of all the transistors are computed using the circuit knowledge of the topology. The gate-to-source voltages (V_{GS}) are determined from I_D, (W/L), $(1+\lambda V_{DS})$, and V_t of the transistors using the following expression

$$V_{GS} = V_t + \left[\frac{I_D}{k'\left(\frac{W}{L}\right)(1+\lambda V_{DS})} \right]^{\frac{1}{2}}. \qquad (5.2)$$

The value of k' is determined from the technology dependent parameters μ and C'_{ox}. In the subsequent step, from the gate to source voltages of the transistors, the various node voltages (V_n) are determined. In the final step, the V_{DS} and the V_{BS} values of all the transistors are evaluated from the node voltages obtained from the previous step. In the subsequent iterations, the values of the V_{DS} and V_{BS} are used to obtain a more accurate estimate of the operating point. When for each node voltage, the results obtained from two consecutive iterations are very close to each other (10^{-6} V in our program), the loop terminates. We have used Shichman-Hodges model [41] for all computational purposes. Once the operating point is determined, all the small signal parameters are calculated using Eqns.(4.3–4.5), as described in Subsection 4.2 .

5.2 THE SIMPLE OPERATIONAL TRANSCONDUCTANCE AMPLIFIER

The schematic of the simple operational transconductance amplifier (OTA), which is optimized in this work, is shown in Fig.5.2.1. This is one of the simplest CMOS op-amps having moderate gain, and is widely used in a variety of applications, one of them is driving on-chip loads, where minimum area is one of the desirable aspects.

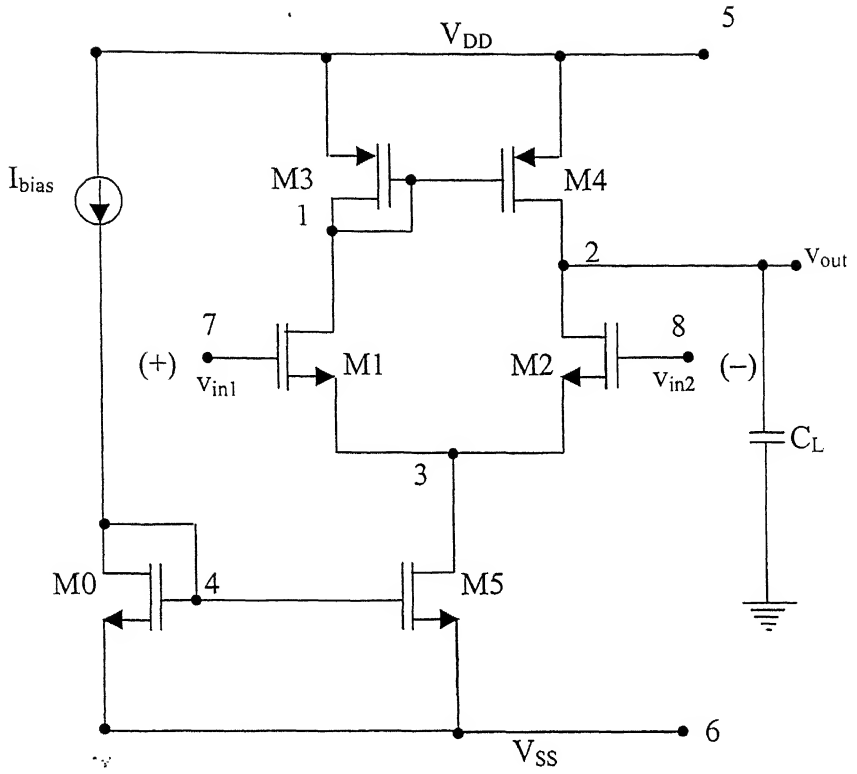


Fig. 5.2.1: The schematic representation of a simple operational transconductance amplifier (OTA).

The design variables for the simple OTA are the lengths (L) and the widths (W) of the individual transistors, and the magnitude of the bias current source I_{bias}. The supply voltages V_{DD} and V_{SS}, and the load capacitance (C_L) are taken to be constants, which are specified by the user. In addition to these, the user is asked to provide the maximum allowable length and width of the transistors to be used by the optimization routine. In order to obtain the independent design variables, which are to be varied during the optimization process, the following simplifications are made.

1. The transistors M1 and M2 are matched differential pair, and, therefore, $(W/L)_1 = (W/L)_2$.
2. The transistors M3 and M4 form the current source acting as active load, and, hence, it can be assumed that $(W/L)_3 = (W/L)_4$.
3. The bias current source consists of transistors M5 and M0, and, therefore, $(W/L)_5 = (W/L)_0$.

The following mathematical equations are used for the computation of the various performance specifications.

1. Gain

The gain of the OTA is given by [46]

$$Gain = g_{m1}\,(r_{o2} \,||\, r_{o4}), \tag{5.2.1}$$

where g_{m1} is the transconductance of M1, and r_{o2} and r_{o4} are the drain-source resistances of M2 and M4 respectively.

2. Unity-gain frequency (UGF)

The unity-gain frequency of the OTA is given by [46]

$$UGF = \frac{g_{m1}}{2\pi\,(C_L + C_{n2})}, \tag{5.2.2}$$

where C_L is the load capacitance, and C_{n2} ($= C_{GD4} + C_{DB4} + C_{GD2} + C_{DB2}$) is the capacitance of node 2 due to the transistors, where C_{GD2} and C_{DB2} are the gate-drain and the drain-body capacitances respectively of M2; and C_{GD4} and C_{DB4} are the gate-drain and the drain-body capacitances respectively of M4.

3. Phase Margin (PM)

The phase margin (*PM*) is given by the following expression [37]

$$PM = 90^0 - \tan^{-1}\left(\frac{UGF}{f_{nd}}\right) + \tan^{-1}\left(\frac{UGF}{2f_{nd}}\right), \tag{5.2.3}$$

where f_{nd} is the non-dominant pole created at node 1 given by

$$= \frac{1}{2\pi\,R_{n1}C_{n1}},$$

where R_{n1} ($\approx 1/g_{m3} = 1/g_{m4}$) is the resistance seen at node 1, g_{m3} and g_{m4} are the transconductances of M3 and M4 respectively, and C_{n1} ($= C_{GD1} + C_{DB1} + C_{DB3} + C_{GS3} +$

$(C_{GS4} + C_{GD4})$ is the capacitance of node 1 due to the transistors connected to it, where C_{GD1} and C_{DB1} are the gate-drain and the drain-body capacitances respectively of M1; C_{GS3} and C_{DB3} are the gate-source and the drain-body capacitances respectively of M3; and C_{GS4} and C_{GD4} are the gate-source and the gate-drain capacitances respectively of M4.

4. *Slew Rate (SR)*

The slew rate (*SR*) is given by

$$SR = \frac{I_{D5}}{(C_L + C_{n2})}, \tag{5.2.4}$$

where I_{D5} is the DC current through M5, which is not equal to I_{bias} due to the greater drain-source voltage across M5 than that across M0.

5. *Input Common Mode Range (CMR)*

The input common-mode range is the range of common-mode input voltages in which the amplifier continues to operate properly [41]. In other words, if a common-mode signal is applied to the gates of M1 and M2, there exist a maximum and a minimum voltage, beyond which the transistors fail to stay in the saturation region. The minimum allowable input voltage (v_{IMIN}) is given by

$$v_{IMIN} = V_{GS1} + V_{GS5} - V_{tno} + V_{SS}. \tag{5.2.5}$$

which puts M5 at the onset of saturation. The maximum allowable input voltage (v_{IMAX}) occurs when the input is pulled towards V_{DD}, and M1 and M2 go into the linear region. This occurs when

$$V_{DS1} = V_{GS1} - V_{tn1} \rightarrow V_{D1} = V_{G1} - V_{tn1}. \tag{5.2.6}$$

Since $V_{G1} = v_{IMAX}$, hence,

$$v_{IMAX} = V_{DD} - V_{SG3} + V_{tn1}. \tag{5.2.7}$$

It is to be noted that the threshold voltage of M1, i.e., V_{tn1} is calculated taking body effect into account. Thus, the input common-mode range over which all the transistors in the op-amp stay in the saturation region is given by

$$\text{Positive CMR } (CMR+) = v_{IMAX}, \text{ and Negative CMR } (CMR-) = v_{IMIN}. \tag{5.2.8}$$

6. *Output swing*

The output swing is determined from the maximum and the minimum output voltages before any of the transistors is pushed into the linear operating region. The maximum output voltage (V_{outmax}) is given by

$$V_{outmax} = V_{DD} - V_{SG4} + V_{tpo}. \tag{5.2.9}$$

Similarly, the minimum output voltage (V_{outmin}) is given by

$$V_{outmin} = V_{SS} + V_{GS5} - V_{tno} + V_{GS2} - V_{tno}. \tag{5.2.10}$$

7. *Common-Mode Rejection Ratio (CMRR)*

The common-mode rejection ratio (*CMRR*) of the differential amplifier is the ratio of the differential-mode gain to the common-mode gain. The common mode gain is given by [41]

$$A_c = \left| \frac{1}{2 g_{m4} r_{o5}} \right|. \tag{5.2.11}$$

Using Eqn.(5.2.1) for the differential-mode gain, the *CMRR* in dB is given by

$$CMRR = 20 \log |2 g_{m1} g_{m4} (r_{o2} \| r_{o4}) r_{o5}|. \tag{5.2.12}$$

8. *Power Supply Rejection Ratio (PSRR)*[1]

One of the most important characteristics of analog circuits is their insensitivity to the spikes present in the power supply lines, expressed by the power supply rejection ratio (*PSRR*). For a dual power supply, the *PSRR* is defined for both the positive supply V_{DD} ($PSRR_{DD}$) and the negative power supply V_{SS} ($PSRR_{SS}$). In analogy to the definition of *CMRR*, $PSRR_{DD}$ and $PSRR_{SS}$ can be defined as [46]

$$PSRR_{DD} = \frac{A_v}{A_{DD}} = \frac{v_{out}}{v_{in}} \frac{v_{DD}}{v_{out}} = \frac{v_{DD}}{v_{in}}, \text{ and} \tag{5.2.13}$$

$$PSRR_{SS} = \frac{A_v}{A_{SS}} = \frac{v_{out}}{v_{in}} \frac{v_{SS}}{v_{out}} = \frac{v_{SS}}{v_{in}}, \tag{5.2.14}$$

where v_{DD} and v_{SS} are the spurious ac signals generated by the clocks (from digital parts or from switched-capacitor circuits), by output drivers, etc.[46], on the power supply lines V_{DD} and V_{SS} respectively. In practice, large capacitors are connected between the power supply lines and the ground to bypass the voltage spikes in order to eliminate their effects on the signal path. In this analysis, it is assumed that such capacitors are not present, and the parasitic capacitances obtained from the layout of the circuit are taken into consideration.

It is possible to obtain high values of *PSRR* at low frequencies, however, at very high frequencies, the response deteriorates [46]. It is to be noted that it is only at the high frequency of operation, that a high value of *PSRR* is required. The spikes on the power supply lines are of short durations and thus contain very high frequency components. As a consequence, a high *PSRR* at only intermediate and high frequencies is required. Therefore, in this analysis we have considered the computation of the *PSRR* at a frequency higher than the bandwidth of the circuit.

At intermediate and higher frequencies, all the small capacitances of the transistors and that of the interconnections come into consideration. The most important ones are the

[1] The Power-Supply Rejection Ratio (PSRR) is of tremendous significance in the mixed-signal design. A detailed explanation and analysis can be found in [46].

coupling capacitances between the supply line V_{DD} and the output [46]. These are referred to as C_{n1DD} and C_{n2DD} between the nodes 1 and 2 and the supply line V_{DD} respectively. They can be calculated from the transistor geometry and the interconnections from the layout of the circuit, and are not known exactly before the design is transformed into a layout. The $PSRR_{DD}$ at a frequency f higher than the bandwidth is given by [46]

$$PSRR_{DD} = \frac{g_{m1}}{g_{o24} + 2\pi f(C_{n1DD} + C_{n2DD})}, \tag{5.2.15}$$

where $g_{o24} = g_{o2} + g_{o4}$, with g_{o2} and g_{o4} being the drain conductances of M2 and M4 respectively. At high frequencies, for the calculation of $PSRR_{SS}$, the most important coupling capacitance is between node 3 and supply line V_{SS}, referred to as Cn_{3SS}. The expression for $PSRR_{SS}$ at a frequency f as mentioned earlier is given by [46]

$$PSRR_{SS} = \frac{g_{m1}}{g_{o5} + 2\pi f C_{n3SS}} \frac{1}{\frac{\Delta g_{m1}}{2g_{m1}} + \frac{\Delta g_{mb1}}{2g_{mb1}}}, \tag{5.2.16}$$

where g_{o5} is the drain-conductance of M5, and Δg_{m1} and Δg_{mb1} are the mismatches in the transconductances and the body transconductances of M1 and M2 respectively. In this work, this mismatch is assumed to be 10 percent for computational purpose.

9. Noise

Each transistor contributes white noise and $1/f$ noise. The equivalent rms input noise voltage of a MOS transistor at any frequency f is given by [46]

$$\overline{dv_n^2(f)} = \frac{8kT}{3}\frac{1}{g_m}df + \frac{K_F}{WL}\frac{df}{f}, \tag{5.2.17}$$

where $KF = KF_F / C_{ox}'^2$, with KF_F being a technology dependent parameter. In order to calculate the noise performance at low frequencies, all the capacitances are omitted. The different noise voltage sources present in an individual transistor can be clubbed together to one equivalent input noise voltage ($\overline{dv_{nie}^2}$) by using the following expression [46]

$$\overline{dv_{nic}^2} = \sum_{i=1}^{n} \overline{dv_{ni}^2} \left(\frac{A_{vni}}{A_{v0}} \right)^2, \tag{5.2.18}$$

where $\overline{dv_{ni}^2}$ is the equivalent noise of the i^{th} transistor referred to its input, A_{vni} is the gain of the i^{th} transistor from its input to the output, and A_{v0} is the overall gain of the circuit. It is observed that the noise due to M1, M2, M3, and M4 dominate the equivalent input noise. Thus, it can be approximately given by the following expression [46]

$$\overline{dv_{nic}^2} \approx 2 \left[\overline{dv_{n1}^2} + \overline{dv_{n3}^2} \left(\frac{g_{m3}}{g_{m1}} \right)^2 \right]. \tag{5.2.19}$$

10. Power dissipation (PD)

The total power dissipation (PD) is given by

$$PD = (V_{DD} - V_{SS})(I_{bias} + I_{D5}). \tag{5.2.20}$$

11. Area

The total area is taken to be equal to the sum of the active areas taken by the individual transistors, and is given by

$$AREA = \sum_{i=0}^{5} W_i \times L_i . \tag{5.2.21}$$

The initial values of the design variables are obtained from a separate routine, which uses the values of the performance specifications specified by the user, the assumptions based on the circuit knowledge, and the simplified analytical equations. The square-law model [42] is used to get a quick estimate of the design variables. The algorithm used to determine the initial values of the design variables is as follows.

1. The initial value of the bias current I_{bias} is determined using Eqn.(5.2.4) from the specified values of the slew rate (SR) and the load capacitance (C_L).

2. The aspect ratios of M1 and M2 are equal, and are calculated using the following equation, by assuming a suitable value of $(V_{GS} - V_{tno})$ (typically 0.2 V – 0.5 V)

$$\left(\frac{W}{L}\right)_1 = \frac{I_{bias}}{k_n'(V_{GS} - V_{tno})^2}. \tag{5.2.22}$$

3. The transconductance of M1 (i.e., g_{m1}) is calculated using the obtained values of I_{bias} and the aspect ratio. Assuming that the output resistance of M2 is equal to that of M4, i.e., $r_{o2} = r_{o4} = r_o$, it is determined using the specified value of the gain and the obtained value of g_{m1} by using Eqn.(5.2.1).

4. The length of M1 (i.e., L1) is determined from the channel length modulation parameter (λ) by using Eqn. (4.2.2), which is obtained from the output resistance, and the dc bias current though it (I_{D1}) from the following expression

$$\lambda = \frac{1}{r_o I_{D1}}. \tag{5.2.23}$$

5. The aspect ratio of M3 is determined by assuming a suitable value of $(V_{SG} - |V_{tpo}|)$ (typically 0.2 V – 0.5 V), and the current flowing through it, and, consequently, the value of g_{m3} is obtained. The length of M3 (i.e., L3) is calculated from its output resistance r_{o3} (= r_{o4} = r_o) and the dc bias current through it.

6. The value of the common-mode gain A_C is determined from the specifications of the gain and the CMRR. The value of the output resistance of M5 (i.e., r_{o5}) is obtained from Eqn.(5.2.11) using the value of g_{m4} (= g_{m3}) obtained from the previous step.

7. The length of M5 (i.e., L5) is obtained from its output resistance and the current through it. By choosing a suitable value of $(V_{GS} - V_t)$ (typically 0.5 V – 1.5 V), the aspect ratio, and, then, the width of M5 (i.e., W5) is determined.

8. The widths of M1 (i.e., W1) and M3 (i.e., W3) are obtained from the calculated values of their respective aspect ratios and lengths.

These initial values of the design variables are supplied to the optimization algorithm in order to obtain the solution to the objective function, which is formulated taking into account the entire set of performance specifications by forming the membership functions for each of them and assigning weights. During each of the iterations, the DC operating point is determined using the algorithm described earlier, and the small signal parameters are calculated. The performances of the circuit are computed using Eqns.(5.2.1–5.2.19).

The output results obtained from the design for a set of performance specifications along with the results obtained from SPICE simulation are listed in Table 5.2.1. The supply voltages V_{DD} and V_{SS}, and the value of the load capacitance (C_L) are taken to be ± 5 V and 10 pF respectively. The *PSRR* and the RMS noise are calculated at frequencies of 100 kHz and 1 kHz respectively. The values of all the parasitic coupling capacitances required for the calculation of the *PSRR* are taken to be 0.2 pF. All these constants are used for the design obtained here, however, some other values can also be supplied to the optimization algorithm as per the users' choice. The parameters obtained from the design and subsequently used for SPICE simulation are presented in Table 5.2.2. The variation of the gain and the phase of the amplifier as a function of frequency, as obtained from SPICE simulation using the circuit parameters obtained from the design of the simple OTA are presented in Figs.5.2.2 and 5.2.3 respectively. From the gain versus frequency plot obtained from SPICE simulation, the values of the gain and the unity-gain frequency of the designed op-amp are determined, and are shown in Table 5.2.1. The values obtained from SPICE simulation are in accordance with those predicted by our optimizer.

In order to determine the common-mode range of the amplifier from SPICE simulation, a common-mode signal is applied to M1 and M2, with their gates tied together. The drain-source and the gate-source voltages across M1 and the current through it as a function of the common-mode signal are plotted in Figs.5.2.4 and 5.2.5 respectively. The positive common-mode range voltage (CMR+) is set by the point where M1 enters the linear region (i.e., $V_{DS1} \leq V_{GS1} - V_{t1}$), which is shown in Fig.5.2.4. It is important to note that the threshold voltage of M1 continuously changes with the input voltage due to body effect, and, therefore, the exact value of the applied common-mode voltage at which M1

enters the linear region is difficult to calculate analytically. However, by substituting the value of the threshold voltage of M1 calculated at the input voltage where $V_{DS1} = V_{GS1}$ (i.e., V_{BS1} = input common-mode voltage – V_{GS1}), an approximate value of CMR+ can be determined. The value obtained from this technique is less than the exact value of CMR+, since the value of $(V_{GS} - V_t)$ decreases with the input common-mode voltage.

Table 5.2.1

The results obtained from the design of the simple OTA for a particular set of performance specifications along with those obtained from SPICE simulation.

Sl. No.	*Symbol*	*Type*	*Weight*	*Target value*	*Value obtained from design*	*Value obtained from SPICE simulation*
1.	*Gain* (dB)	≥	0.1	45	51.15	51.49
2.	*UGF* (MHz)	≥	0.1	10	14.08	14.2
3.	*PM* (degree)	≥	0.05	80	82.11	85.5
4.	*SR* (V/μs)	≥	0.1	5	4.98	5.19
5.	*CMR+* (V)	≥	0.05	3.0	4.55	4.3
6.	*CMR–* (V)	≤	0.05	–3.0	-3.47	-3.7
7.	V_{outmax} (V)	≥	0.05	3.5	4.277	4.366
8.	V_{outmin} (V)	≤	0.05	-3.5	-4.439	-4.381
9.	*CMRR* (dB)	≥	0.1	60	89.617	89.915
10.	$PSRR_{DD}$ (dB)	≥	0.05	50	50.507	50.00
11.	$PSRR_{SS}$ (dB)	≥	0.05	50	54.537	55.21
12.	*RMS noise* (nV/√Hz)	≤	0.05	10	4.334	*
13.	*PD* (mW)	≤	0.1	5	0.921	0.879
14.	*AREA* (μ^2)	≤	0.1	200	145.877	*

* SPICE3 in the computer center does support calculation of these parameters.

Table 5.2.2

The set of design variables obtained from the design of the simple OTA for a set of performance specifications, and subsequently used for SPICE simulations.

Sl. No.	*Design variable*	*Obtained value*
1.	W/L of M1, M2 (μm/μm)	70.453/0.824
2.	W/L of M3, M4 (μm/μm)	1.967/1.232
3.	W/L of M0, M5 (μm/μm)	5.728/2.108
4.	Bias current (I_{bias}) (μA)	41.807

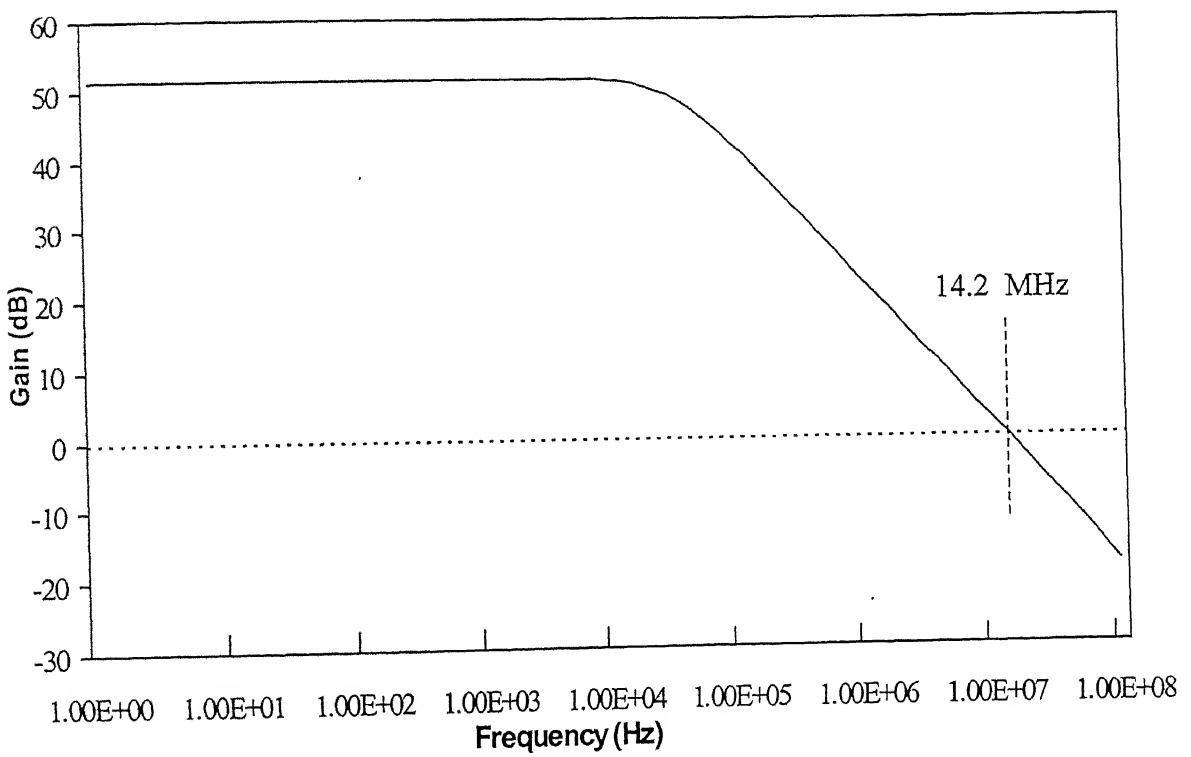


Fig.5.2.2: The gain versus frequency plot for the designed simple OTA, as obtained from SPICE simulation. The unity-gain frequency of 14.2 MHz is also shown in the figure.

Figure 5.2.5 shows the current through M1 versus the voltage applied to its gate. The negative common-mode range (CMR–) is shown in the figure as the point where the

Table 5.2.2

The set of design variables obtained from the design of the simple OTA for a set of performance specifications, and subsequently used for SPICE simulations.

Sl. No.	*Design variable*	*Obtained value*
1.	W/L of M1, M2 (μm/μm)	70.453/0.824
2.	W/L of M3, M4 (μm/μm)	1.967/1.232
3.	W/L of M0, M5 (μm/μm)	5.728/2.108
4.	Bias current (I_{bias}) (μA)	41.807

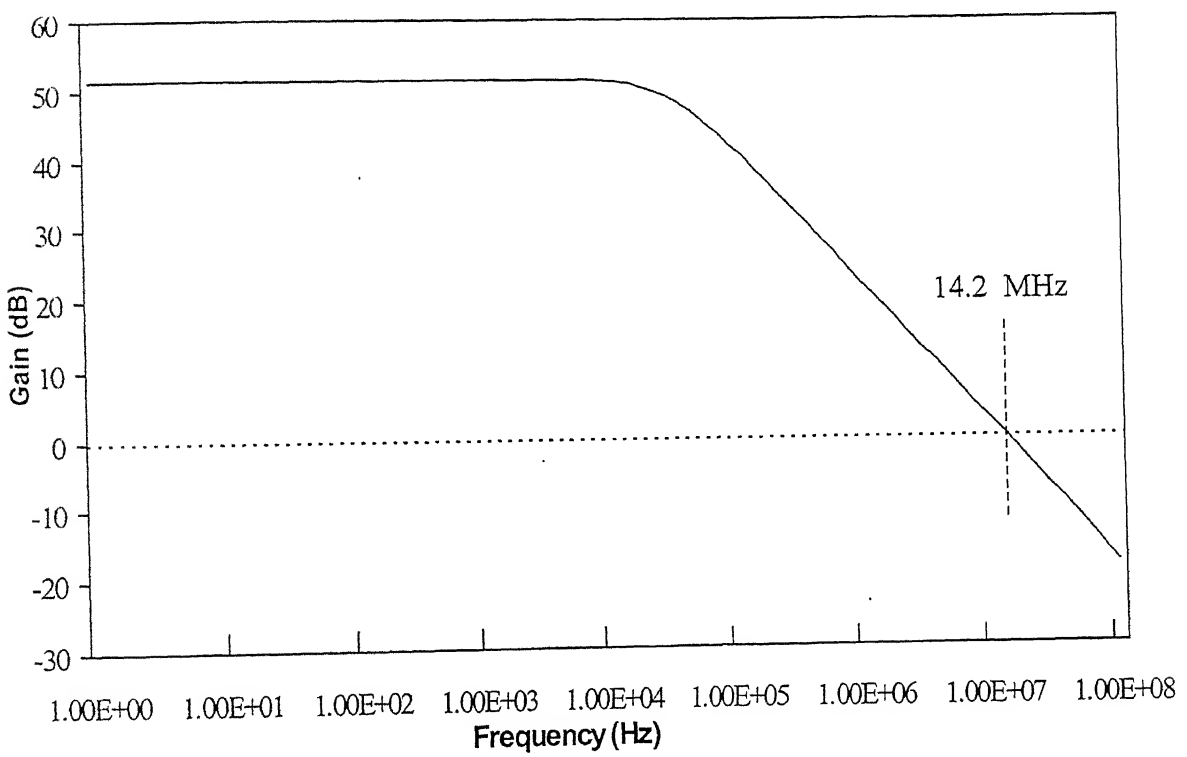


Fig.5.2.2: The gain versus frequency plot for the designed simple OTA, as obtained from SPICE simulation. The unity-gain frequency of 14.2 MHz is also shown in the figure.

Figure 5.2.5 shows the current through M1 versus the voltage applied to its gate. The negative common-mode range (CMR–) is shown in the figure as the point where the

current through M1 starts to decrease and approach zero, and the current through M5 also starts to drop, indicating its entrance to the linear region.

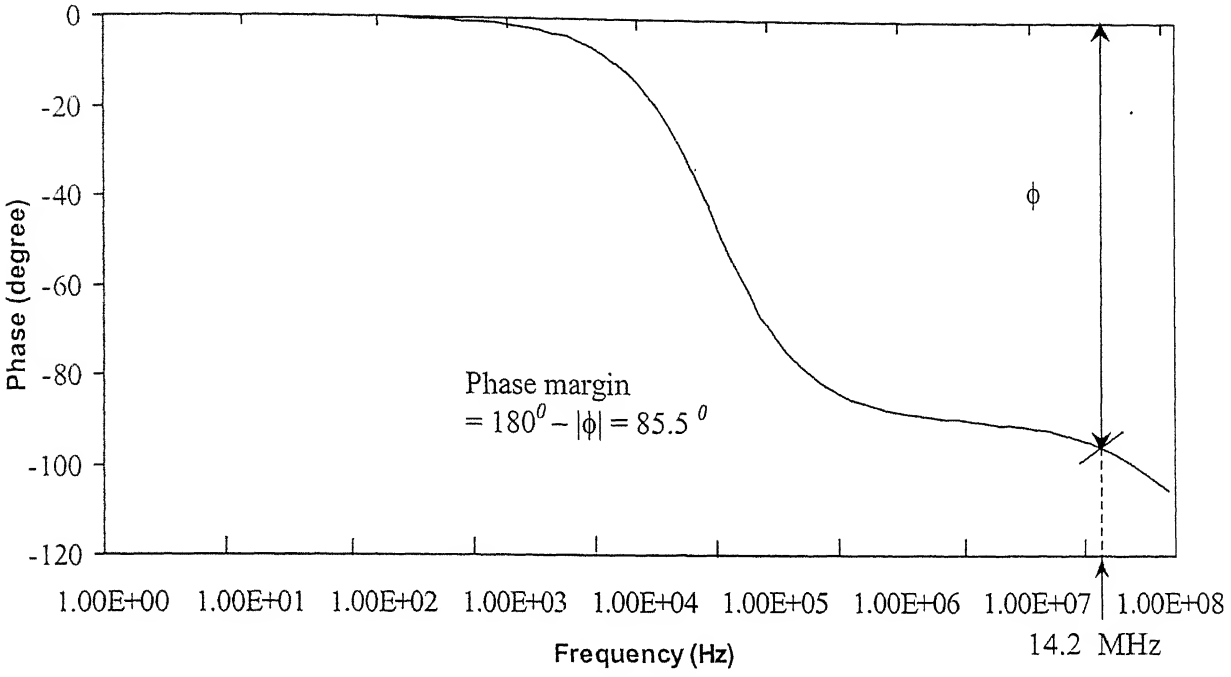


Fig.5.2.3: The phase versus frequency plot for the designed simple OTA, as obtained from SPICE simulation. The phase margin of 85.5^0 at the unity-gain frequency of 14.2 MHz is also shown in the figure.

In order to get the values of *PSRR* due to V_{DD} and V_{SS} from SPICE simulation, capacitors are connected between the corresponding nodes with their values equal to those of the parasitic coupling capacitances, and an ac signal is superimposed with the DC supply voltage. The gain of the amplifier due to the ac power supply voltage is determined, and with the knowledge of the differential gain, the value of PSRR is obtained. The results obtained from our optimizer closely match with those obtained from SPICE simulation for all the performance specifications. However, the value of the *RMS noise*, as referred to the input of the op-amp, could not be verified, since SPICE3 does not support its calculation.

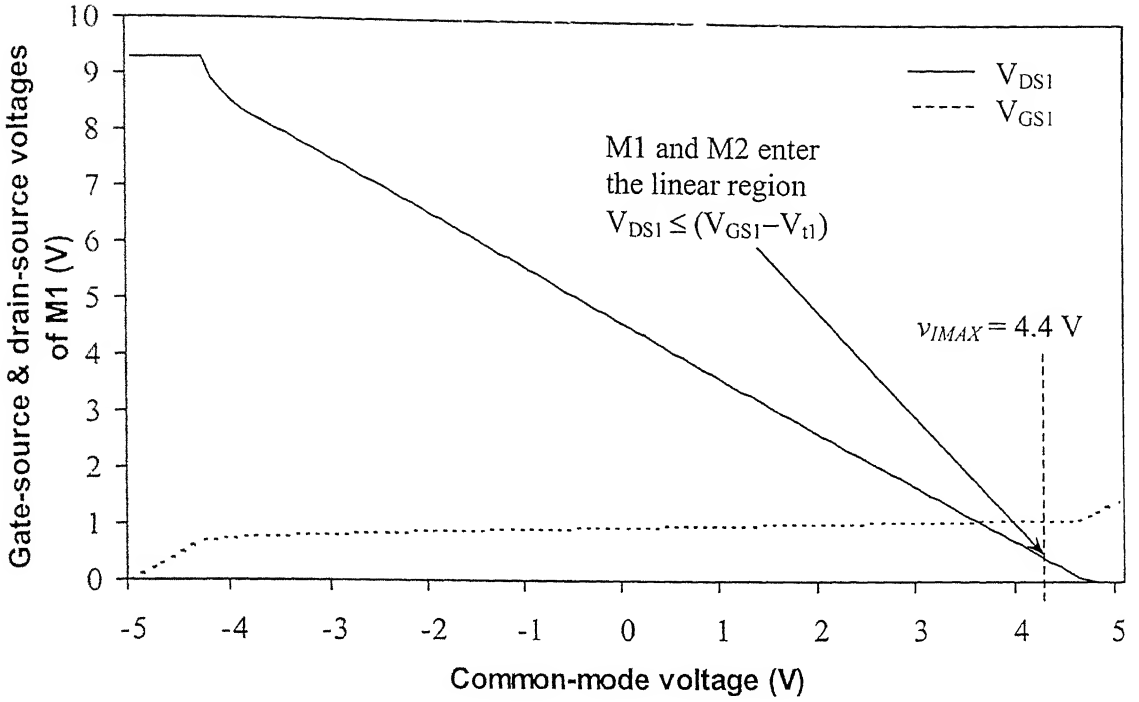


Fig. 5.2.4: The variation of the gate-source voltage (V_{GS1}) and the drain-source voltage (V_{DS1}) of M1 with the applied common-mode input voltage, as obtained from SPICE simulation.

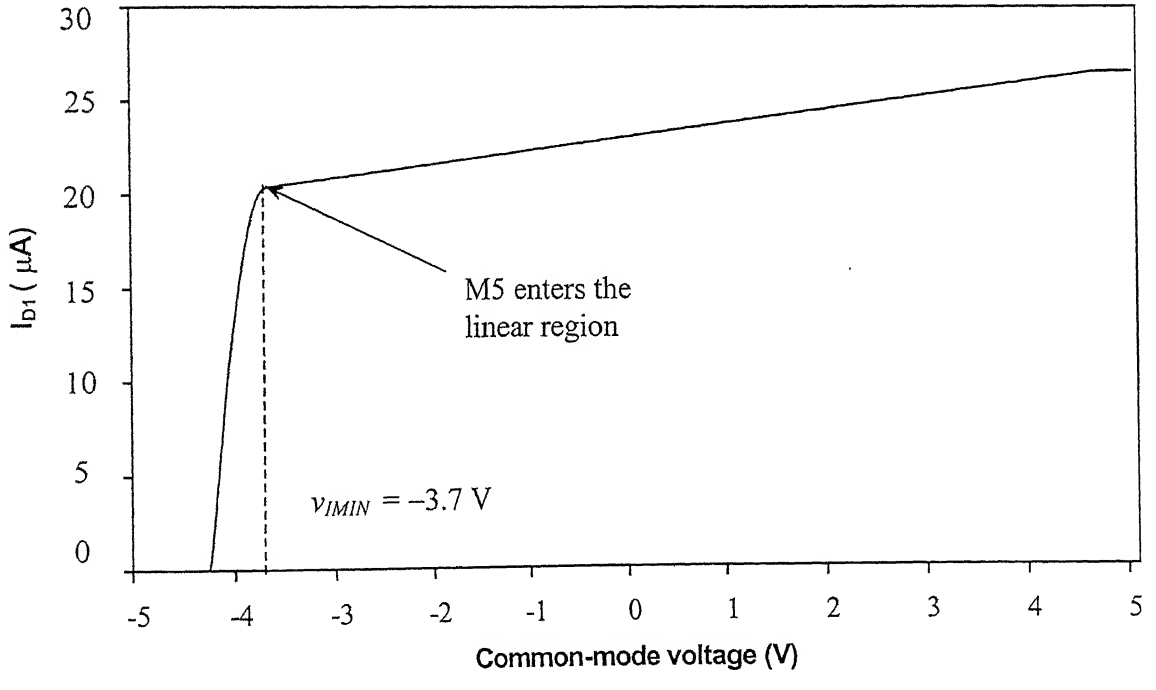


Fig.5.2.5: The variation of the drain current of M1 with the common-mode input voltage to determine the negative common-mode range, as obtained from SPICE simulation.

5.3 THE MILLER COMPENSATED OPERATIONAL TRANSCONDUCTANCE AMPLIFIER (BASIC TWO-STAGE OP-AMP)

The schematic of the basic two-stage CMOS op-amp is shown in Fig.5.3.1. It consists of two stages, the first of which is a differential stage with the PMOS input devices M1 and M2, and the NMOS current mirror M3 and M4, also acting as the active load. The second stage is a simple CMOS inverter with M6 as the driver and M5 as the active load. The output of the gain stage is connected to its input through the compensating capacitor C_C. Since C_C actually acts as the Miller capacitance of that stage, the op-amp is called a Miller compensated OTA. In this work, this circuit topology is selected for optimization, considering the suitability of its fabrication with an *n*-well technology, which our technology file is based upon.

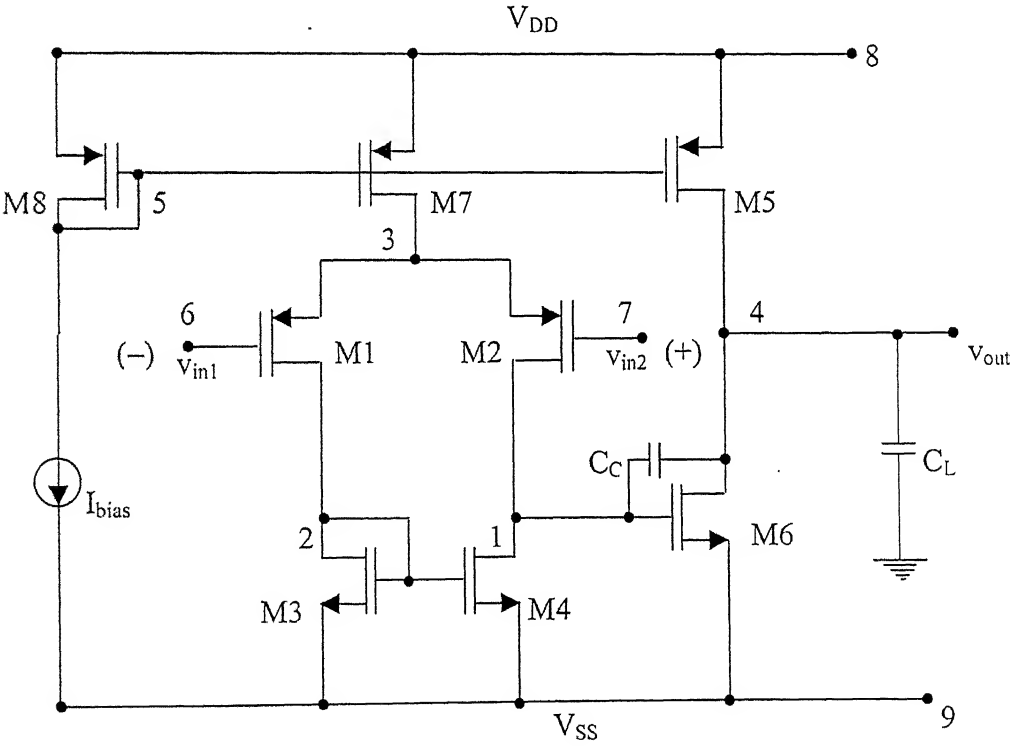


Fig.5.3.1: Schematic of the Miller compensated CMOS operational transconductance amplifier (OTA).

The design variables for this circuit are the lengths (L) and widths (W) of the transistors, the bias current source I_{bias}, and C_C. The supply voltages V_{DD} and V_{SS}, and the load capacitance (C_L) are taken as constants to be specified by the user. In order to get the independent design variables, which are to be varied during the optimization process, the following assumptions are made.

1. The input transistors M1 and M2 are matched differential pairs, and, therefore, $(W/L)_1 = (W/L)_2$.
2. The transistors M3 and M4 are current sources also acting as active load, and, hence, $(W/L)_3 = (W/L)_4$.
3. The bias current source consists of transistors M7 and M8, and, therefore, $(W/L)_7 = (W/L)_8$.

4. The PMOS load transistor M5 of the gain stage has independent W/L ratio, and it determines the current though that stage. The aspect ratio of the driver M6 depends on the current through the second stage. The drain-source voltage of M6 is fixed (as the DC output node voltage is taken to be equal to zero), and its gate-source voltage is obtained from the drain-source voltage of M4, and, therefore, only its length L6 is taken to be an independent variable.

Hence, the independent design variables of the above circuit topology are W1, L1, W3, L3, W7, L7, W5, L5, L6, C_C, and I_{bias}. The following mathematical equations are used for the computation of the various performance specifications.

1. Gain

The gain of the basic two-stage (BTS) op-amp is the product of the gain of the first stage and that of the second stage. The first stage is a simple OTA as described in Subsection 5.2, the gain of which is given by

$$Gain1 = -g_{m1}(r_{o2} \| r_{o4}). \tag{5.3.1}$$

The second stage is an inverter, the gain of which is given by

$$Gain2 = -g_{m6}(r_{o5} \| r_{o6}). \tag{5.3.2}$$

Thus, the overall gain can be expressed as

$$Gain = Gain1 \times Gain2 = g_{m1}g_{m6}(r_{o2} \| r_{o4})(r_{o5} \| r_{o6}). \tag{5.3.3}$$

2. *Unity-gain frequency (UGF)*

The exact expression for the unity-gain frequency is given by [46]

$$UGF = \frac{g_{m1}}{2\pi C_C}\frac{1}{1+\varepsilon_{UGF}}, \tag{5.3.4}$$

where $\varepsilon_{UGF} = \frac{(g_{o5}+g_{o6})}{g_{m6}}\left(1+\frac{C_{n1}}{C_C}\right)+\frac{(g_{o2}+g_{o4})}{g_{m6}}\left(1+\frac{C_L^{'}}{C_C}\right)$,

where C_{n1} (= C_{GS6} + C_{GD2} + C_{DB2} + C_{GD4} + C_{DB4}) is the transistor capacitances of node 1, $C_L^{'}$ = C_L + C_{n4}, with C_{n4} (= C_{GD5} + C_{DB5} + C_{DB6} + C_{GD6}) being the transistor capacitances of node 4, where C_{GD2} and C_{DB2} are the gate-drain and the drain-body capacitances of M2 respectively, C_{GD4} and C_{DB4} are the gate-drain and the drain-body capacitances of M4 respectively, C_{GD5} and C_{DB5} are the gate-drain and the drain-body capacitances of M5 respectively, and C_{GS6}, C_{GD6}, and C_{DB6} are the gate–source, the gate-drain and the drain-body capacitances of M6 respectively.

3. *Phase Margin (PM)*

The phase margin (*PM*) is given by the following expression [46]

$$PM = 90^0 - \tan^{-1}\left(\frac{UGF}{f_{nd}}\right), \tag{5.3.5}$$

where f_{nd} = non-dominant pole created at node 4 given by [46]

$$= \frac{g_{m6}}{2\pi C_L^{'}}\left(\frac{1+\varepsilon_{UGF}}{1+\frac{C_{n1}}{C_C}+\frac{C_{n1}}{C_L^{'}}}\right). \tag{5.3.6}$$

4. Slew Rate (SR)

The slew rate (SR) is given by

$$SR = \frac{I_{bias}}{C_C}. \tag{5.3.7}$$

5. Input Common-Mode Range (CMR)

The maximum allowable input voltage is given by the sum of the voltages from the input of the diff-amp stage to V_{DD} when the input is at v_{IMAX} and M7 gets to the verge of saturation, and can be represented by the following equation

$$v_{IMAX} = V_{DD} - |V_{GS1}| - |V_{GS7}| + |V_{tpo}|. \tag{5.3.8}$$

The minimum allowable input voltage occurs when the input is pulled towards V_{SS}, and M1 and M2 just enter the linear region. This occurs when

$$|V_{DS1}| = |V_{GS1}| - |V_{tpo}| \rightarrow V_{D1} = V_{G1} - |V_{tpo}|. \tag{5.3.9}$$

Since $V_{G1} = v_{IMIN}$, hence

$$v_{IMIN} = V_{SS} + V_{GS3} - V_{tno}. \tag{5.3.10}$$

The values of the positive common-mode range (CMR+) and the negative common-mode range (CMR–) can be obtained using Eqn.(5.2.8).

6. Output swing

The output swing is determined from the maximum and the minimum output voltages such that all the transistors stay in the saturation region. The maximum output voltage (V_{outmax}) is given by

$$V_{outmax} = V_{DD} - V_{DSAT5} = V_{DD} - |V_{GS5}| + |V_{tpo}|. \tag{5.3.11}$$

The minimum output voltage (V_{outmin}) is given by

$$V_{outmin} = V_{SS} + V_{DSAT6} = V_{SS} + V_{GS6} - V_{tno}. \tag{5.3.12}$$

7. *Common-Mode Rejection Ratio (CMRR)*

The common-mode rejection ratio (*CMRR*) of the BTS op-amp is due to the first stage only. The second stage amplifies both the common-mode and the differential-mode signals. Thus, the CMRR in dB is given by

$$CMRR = 20 \log |2 g_{m1} g_{m4} (r_{o2} \| r_{o4}) r_{o7}|. \tag{5.3.13}$$

8. *Power Supply Rejection Ratio (PSRR)*

The power supply rejection ratio due to V_{DD} ($PSRR_{DD}$) is determined from the contribution of two different currents through M7 and M5, both providing a path from V_{DD} to v_{out} The $PSRR_{DD}$ at any frequency f, which is higher than the band width of the circuit is given by [46]

$$PSRR_{DD} = \frac{\dfrac{v_{out}}{v_{in}}}{\dfrac{v_{out7} + v_{out5}}{v_{DD}}} = \frac{g_{m1}}{(g_{o7} + 2\pi f C_{n3DD})\dfrac{\Delta g_{m1}}{2 g_{m1}} + 2\pi f C_C \left(\dfrac{g_{o5} + 2\pi f C_{n4DD}}{g_{m6}} \right)}, \tag{5.3.14}$$

where v_{out5} and v_{out7} are the voltages developed at the output due to the passage of the spurious ac signal v_{DD} through M5 and M7 respectively, and C_{n3DD} and C_{n4DD} are the parasitic coupling capacitances between nodes 3 and 4 to the supply line V_{DD} respectively, determined partly from the transistors' geometry and partly from the layout of the circuit.

The power supply rejection ratio due to V_{SS} ($PSRR_{SS}$) consists of three components, due to the currents through M3, M4, and M6. The $PSRR_{SS}$ at a frequency f, as mentioned earlier is given by [46]

$$PSRR_{SS} = \frac{\frac{v_{out}}{v_{in}}}{\frac{v_{out3} + v_{out4} + v_{out6}}{v_{SS}}},$$

$$= \frac{g_{m1}}{(g_{o2} + g_{o4}) + 2\pi f(C_{n23} + C_{n1SS} + C_C)}, \tag{5.3.15}$$

where v_{out3}, v_{out4}, and v_{out5} are the voltages appearing at the output node due to the spurious ac signal v_{SS} present on the power supply line V_{SS} through M3, M4, and M6 respectively, C_{n23} is the parasitic coupling capacitance between nodes 2 and 3, and C_{n1SS} is the parasitic coupling capacitance between node 1 and V_{SS}.

9. Noise

The equivalent input noise is dominated by the noise due to M1/M2 and M3/M4, and is the same as that for a simple OTA. Thus, the equivalent input noise can be obtained using Eqn.(5.2.18).

10. Power dissipation (PD)

The total power dissipated (*PD*) is given by

$$PD = (V_{DD} - V_{SS}) \times (I_{bias} + I_{D7} + I_{D5}). \tag{5.3.16}$$

11. Area

The total area is given by the sum of the active area taken by the individual transistors, and is given by

$$AREA = \sum_{i=1}^{8} W_i \times L_i. \tag{5.3.17}$$

In order to determine the initial values of the design variables, an algorithm is developed and implemented in this work. The algorithm deduces the values of the design

variables from the performance specifications specified by the user, and the circuit knowledge using the simple square-law current equation of the MOS transistors. The algorithm works as described below.

1. The gain of the BTS op-amp is obtained by taking the product of the gain of the first stage and that of the second stage. Assuming that both the stages provide approximately equal gain, the value of the gain of each stage can be obtained.

2. The value of C_C is taken to be the same as that of the load capacitance C_L. From the value of the slew rate (SR), the bias current I_{bias} is determined using Eqn.(5.3.7).

3. Using the specified values of CMRR, the gain of the first stage, and the bias current, the lengths and widths of the transistors of the first stage can be obtained as described earlier in Subsection 5.2 for the design of the simple OTA.

4. A suitable ratio of the current flowing through the first stage and the second stage (typically 1–4) is assumed in order to calculate the current through the latter stage. From the value of $(V_{GS} - V_t)$, as obtained from the first stage, the aspect ratios of M5 and M6 are obtained.

5. The transconductance of M6 (i.e., g_{m6}) is determined by using the value of the current through it, the aspect ratio, and the process transconductance parameter. It is assumed that the output resistances of M5 and M6 are equal.

6. In the next step, from the chosen value of the gain of the second stage, and by using the calculated value of g_{m6}, the output resistances of M5 and M6 are obtained using Eqn.(5.3.2), and the lengths of M5 and M6 are determined. Finally, the widths of M5 and M6 are determined from the aspect ratios, as obtained from the previous steps.

These initial values of the design variables are supplied to the optimization algorithm in order to solve the objective function. The formulation procedure of the objective function and the values of the constants parameters supplied to the optimization algorithm are the same as those for the simple OTA, described earlier in Subsection 5.2. The output results obtained from the design for a set of performance specifications along with the results obtained from SPICE simulation are listed in Table 5.3.1. The supply

voltages V_{DD} and V_{SS}, and the value of the load capacitance (C_L) are taken to be ± 5 V and 10 pF respectively. The values of the PSRR and the RMS noise are calculated at frequencies of 100 kHz and 1 kHz respectively. The values of all the parasitic coupling capacitances required for the calculation of the power-supply rejection ratios are taken to be 0.2 pF. The design variables obtained from the optimization program and subsequently used for SPICE simulation are presented in Table 5.3.2.

Table 5.3.1

Results obtained from optimization of the design of the basic two-stage op-amp for a particular set of performance specifications along with those obtained from SPICE simulation.

Sl. No.	*Symbol*	*Type*	*Weight*	*Target value*	*Value obtained from design*	*Value obtained from SPICE simulation*
1.	*Gain* (dB)	≥	0.1	90	93.133	96.71
2.	*UGF* (MHz)	≥	0.1	10	10.155	13.2
3.	*PM* (degree)	≥	0.05	45	46.72	42.3
4.	*SR* (V/μs)	≥	0.1	5	6.084	7.16
5.	*CMR+* (V)	≥	0.05	3	3.409	3.435
6.	*CMR−* (V)	≤	0.05	-3	-4.935	-4.913
7.	V_{outmax} (V)	≥	0.05	3.5	4.712	4.675
8.	V_{outmin} (V)	≤	0.05	-3.5	-4.935	-4.913
9.	*CMRR* (dB)	≥	0.1	90	99.609	103.96
10.	$PSRR_{DD}$ (dB)	≥	0.05	60	119.763	117.38
11.	$PSRR_{SS}$ (dB)	≥	0.05	60	59.998	61.35
12.	*RMS noise* (nV/√Hz)	≤	0.05	10	8.662	*
13.	*PD* (mW)	≤	0.1	5	1.474	1.546
14.	*AREA* (μ^2)	≤	0.1	500	462.874	*

*SPICE3 in the computer center does not support calculation of these parameters

The gain versus frequency and the phase versus frequency plots, as obtained from SPICE simulation are presented in Figs.5.3.2 and 5.3.3 respectively. The values of the gain and the unity-gain frequency obtained from the design using the synthesis tool developed in this work match reasonably well with those obtained from SPICE simulation. Although the designed circuit has a phase margin of 42.3°, it is not stable due to the existence of right half-plane zero along with the two poles. Physically, the zero arises due to C_C, which provides a signal path through it directly connecting the input to the output at high frequencies without any phase inversion. At high frequencies, the second stage presents a load equal to $1/g_{m6}$ to the input stage. The gain of the amplifier at that point is determined by the ratio of g_{m1}/g_{m6}, with the polarity being opposite to that at low frequencies (when the second stage acts as an inverter), thereby changing the negative feedback, which might be present, into a positive feedback.

Table 5.3.2

The design variables obtained from the optimization routine for the BTS op-amp, which are subsequently used for SPICE simulation.

Sl. No.	*Design variable*	*Obtained value*
1.	W/L of M1, M2 (μm/μm)	112.413/0.8
2.	W/L of M3, M4 (μm/μm)	80.21/0.93
3.	W/L of M5, M7, M8 (μm/μm)	17.444/0.989
4.	W/L of M6 (μm/μm)	102.42/0.8
5.	Compensating capacitor C_C (pF)	7.953
6.	Bias current (I_{bias}) (μA)	48.38

The effect of the right half-plane zero can be eliminated by inserting a resistance R_C ($\geq 1/g_{m6}$) in series with C_C. The phase margin improves with it, and the variation of the phase with frequency for different values of R_C, as obtained from SPICE simulation, is plotted in Fig.5.3.3. It is observed that the required value of the nulling resistor R_C in order to eliminate the effect of the right half-plane zero completely is about twice that of the

resistance $1/g_{m6}$. However, if a larger value of R_C is used, the phase variation shows a rise in the phase margin around the frequency ranges of UGF, and then decreases again at higher frequencies.

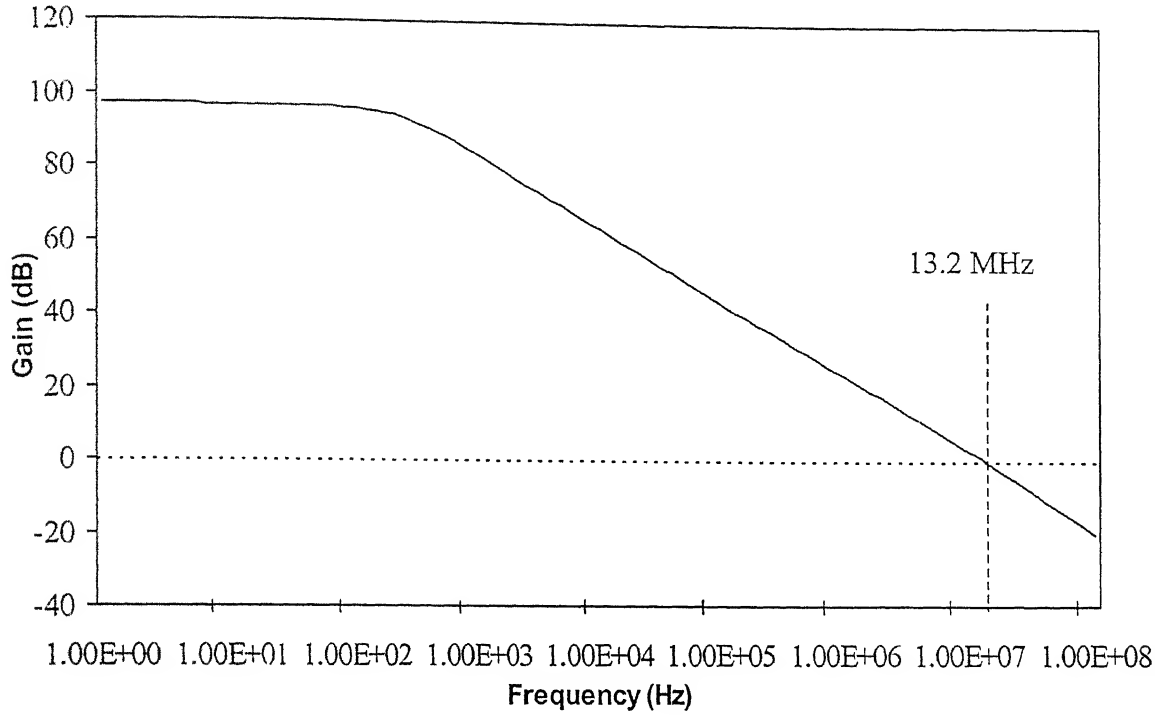


Fig.5.3.2: The gain versus frequency plot for the 2-stage Miller compensated CMOS op-amp obtained using the optimized values of the design variables satisfying the set of performance specifications as shown in Table 5.3.1. The unity-gain cutoff at 13.2 MHz is also shown in the figure.

It is worthwhile to discuss the value of the power supply rejection ratio due to V_{SS} (i.e., $PSRR_{SS}$) obtained from the design routine. At high frequencies, C_C acts as a short circuit. Hence, M6 behaves like a diode connected transistor, offering a load impedance of $1/g_{m6}$, and, thus, the spurious signals on the V_{SS} line reach the output without any attenuation. This is one definite disadvantage of the Miller compensated OTA [46]. It is evident from Table 5.3.1 that all the specifications are fully satisfied, and they match reasonably well with the results obtained from SPICE simulation. The lower value of the phase margin obtained from the simulation results is due to the fact that the existence of the

right half-plane zero is not taken into consideration in the equations used for computation in our optimization routine.

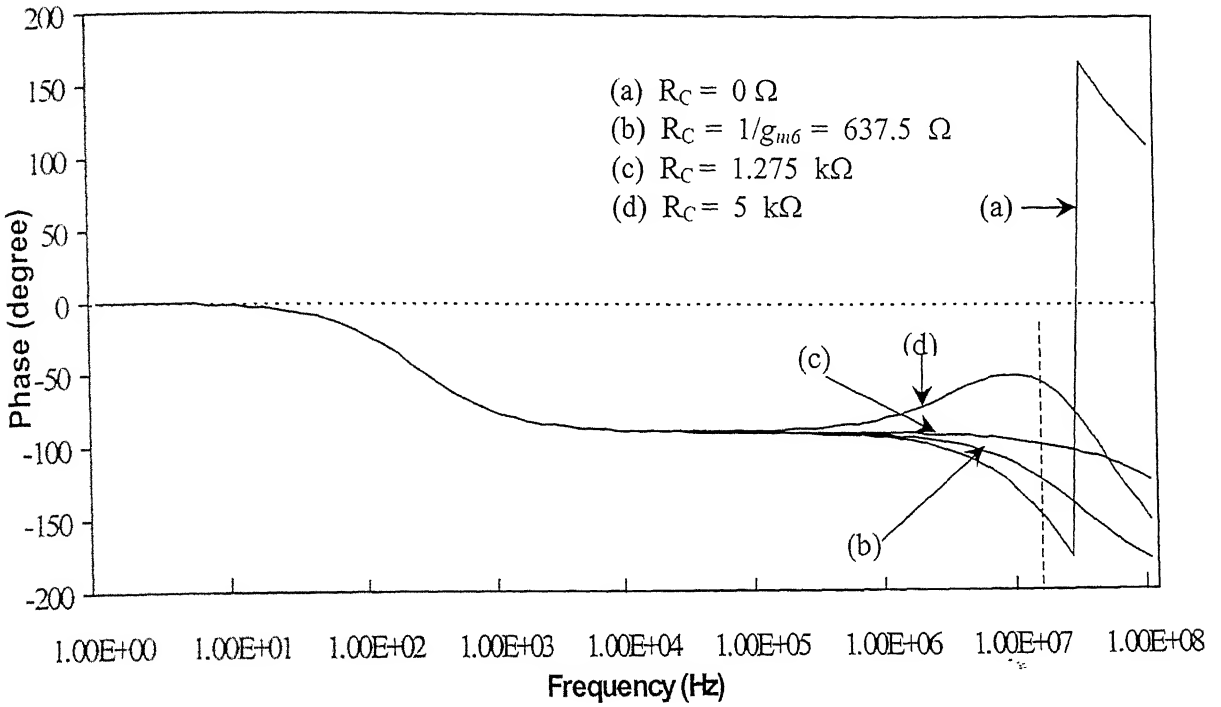


Fig.5.3.3: The phase versus frequency plot for the 2-stage Miller compensated CMOS op-amp obtained from SPICE simulation using the optimized values of the design variables satisfying the given set of performance specifications, and with the values of the nulling resistor R_C (used in series with C_C) of: (a) 0 Ω, (b) $1/g_{m6}$ = 637.5 Ω, (c) 1.275 kΩ, and (d) 5 kΩ.

5.4 THE SYMMETRICAL OPERATIONAL TRANSCONDUCTANCE AMPLIFIER

Both the simple OTA and the Miller compensated OTA consist of input differential pair loaded by a current mirror, and, thus, they are loaded asymmetrically. In order to load both the input transistors equally, a symmetrical OTA is used, the schematic of which is shown in Fig.5.4.1. M1 and M2 form the differential pair, driving the output current in M3 and M4 connected in diode configuration. M3/M5 and M4/M6 form current mirrors with a multiplication factor *B*. The current in M5 is mirrored in M7 and M8. In order to formulate

the objective function from the values of the performance specifications as specified by the designer, and then apply the optimization algorithm developed in this work to solve it, the design parameters are obtained using the following assumptions and circuit knowledge.

1. M1 and M2 are matched pairs, and, therefore, $(W/L)_1 = (W/L)_2$. M3 and M4 have the same aspect ratios in order to load both the input transistors symmetrically, the idea with which the topology is used. The current mirror used for biasing is formed by M9 and M10, and, hence, $(W/L)_9 = (W/L)_{10}$.

2. In order to reduce the number of the circuit parameters (i.e., independent design variables), it is assumed that the gate-source voltages of M3 to M8 are the same, and that they have equal lengths. However, the widths of these transistors are determined from their gate-source voltages, drain-source voltages, and the current flowing through them.

3. The current flowing through M5 and M6 is B times the current through the input transistors and it is considered as an independent design variable.

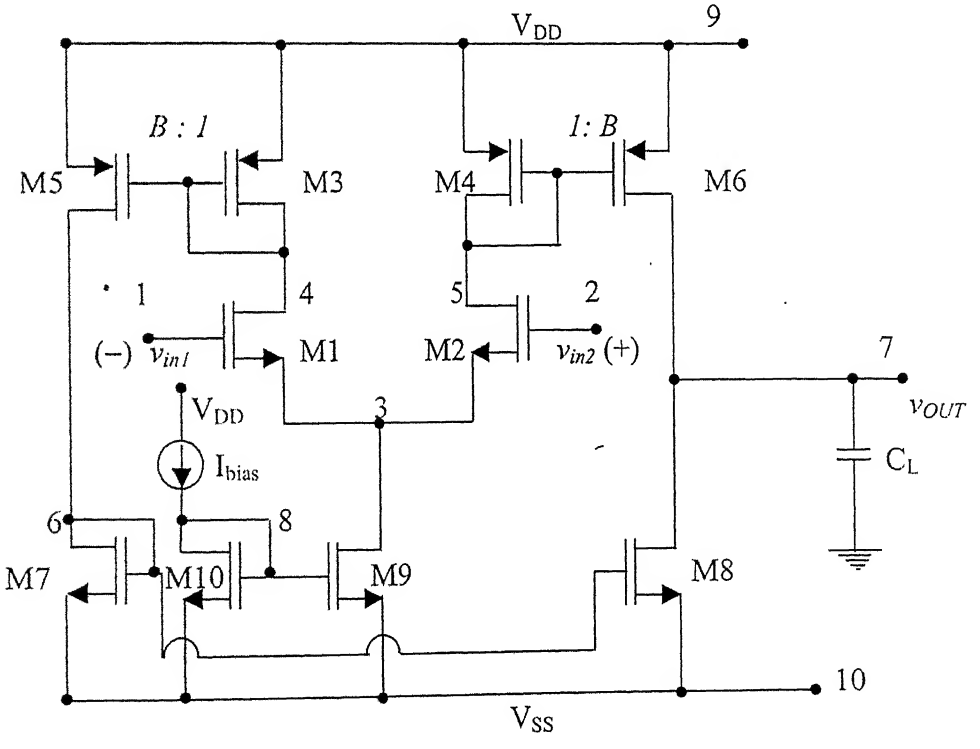


Fig.5.4.1: Schematic of a symmetrical CMOS operational transconductance amplifier.

Therefore, this circuit topology has a set of eight independent parameters, i.e., W1, L1, W3, L3, W9, L9, B, and I_{bias}. The following mathematical equations are used for the computation of the various performance specifications.

1. Gain

The gain of the symmetrical OTA is given by [46]

$$Gain = B\,g_{m1}\,(r_{o6} \| r_{o8}). \tag{5.4.1}$$

2. Unity-gain frequency (UGF)

The unity-gain frequency is given by [46]

$$UGF = \frac{B\,g_{m1}}{2\pi\,(C_L + C_{n7})}, \tag{5.4.2}$$

where C_{n7} (= C_{GD6} + C_{DB6} + C_{GD8} + C_{DB6}) is the capacitance of node 7 due to transistors, where C_{GD6} and C_{DB6} are the gate-drain and the drain-body capacitances of M6 respectively, and C_{GD8} and C_{DB8} are the gate-drain and the drain-body capacitances of M8 respectively.

3. Phase Margin (PM)

The phase margin (*PM*) is given by the following expression [46]

$$PM = 90^0 - \varphi_{n5} + \varphi_{n6}, \tag{5.4.3}$$

where $\varphi_{n5} = \tan^{-1}\left(\frac{UGF}{f_{nd5}}\right)$,

with $f_{nd5} \approx \frac{g_{m4}}{2\pi\,C_{n5}}$,

where C_{n5} (= C_{DB4} + C_{GS4} + C_{GS6} + C_{GD6} + C_{GD2}) is the capacitance at node 5 due to transistors, where C_{GS4} and C_{DB4} are the gate-source and the drain-body capacitances of M4 respectively, and, C_{GD2} is the gate-drain capacitance of M2, and

$$\varphi_{n6} = \tan^{-1}\left(\frac{UGF}{f_{n6}}\right) - \tan^{-1}\left(\frac{UGF}{2f_{n6}}\right),$$

with $f_{nd6} \approx \frac{g_{m7}}{2\pi C_{n6}}$,

where C_{n6} (= C_{DB7} + C_{GS7} + C_{GS8} + C_{GD8} + C_{DB5} + C_{GD5}) is the capacitance at node 6 due to transistors, where C_{GD5} and C_{DB5} are the gate-source and the drain-body capacitances of M5 respectively, and C_{GS7} and C_{DB7} are the gate-source and the drain-body capacitances of M7 respectively.

4. *Slew Rate (SR)*

The slew rate (*SR*) is given by

$$SR = \frac{I_{D5}}{(C_L + C_{n7})}. \tag{5.4.4}$$

5. *Input Common Mode Ranges (CMR+ and CMR−)*

The minimum allowable input voltage is given by summing the voltages from the input of the diff-amp to V_{SS} when the input is at v_{IMIN} and M9 is on the verge of saturation, and can be represented by the following equation

$$v_{IMIN} = V_{GS1} + V_{GS9} - V_{tno} + V_{SS}. \tag{5.4.5}$$

The maximum allowable input voltage occurs when the input is pulled towards V_{DD}, and M1 and M2 enter the linear region. This occurs when

$$V_{DS1} = V_{GS1} - V_{tn1} \rightarrow V_{D1} = V_{G1} - V_{tn1}. \tag{5.4.6}$$

Since $V_{G1} = v_{IMAX}$, hence

$$v_{IMAX} = V_{DD} - V_{SG3} + V_{tn1}. \tag{5.4.7}$$

The input common-mode ranges can thus be obtained by using Eqn.(5.2.8).

6. *Output swing*

The output swing is determined from the maximum and the minimum output voltages before any of the transistors is pushed into the linear operating regime. The maximum output voltage (V_{outmax}) is given by

$$V_{outmax} = V_{DD} - V_{DSAT6} = V_{DD} - |V_{GS6}| + |V_{tpo}|. \tag{5.4.8}$$

The minimum output voltage (V_{outmin}) is given by

$$V_{outmin} = V_{SS} + V_{DSAT8} = V_{SS} + V_{GS8} - V_{tno}. \tag{5.4.9}$$

7. *Common-Mode Rejection Ratio (CMRR)*

The common-mode rejection ratio ($CMRR$) in dB is given by

$$CMRR = 20 \log \left| 2 g_{m1} g_{m4} \left(r_{o2} \| \frac{1}{g_{m4}} \right) r_{o9} \right|. \tag{5.4.10}$$

8. *Power Supply Rejection Ratio (PSRR)*

The $PSRR_{DD}$ at any frequency f, which is higher than the bandwidth of the circuit is given by

$$PSRR_{DD} = \frac{\dfrac{i_{OUT}}{v_{IN}}}{\dfrac{i_{OUT}}{v_{DD}}} = \frac{B g_{m1}}{(g_{o6} + 2\pi f C_{n7DD})}, \tag{5.4.11}$$

where C_{n7DD} is the parasitic coupling capacitance between the output node 7 and the supply voltage V_{DD} line.

The $PSRR_{SS}$ at a frequency f, as mentioned earlier is given by

$$PSRR_{SS} = \frac{\dfrac{i_{OUT}}{v_{IN}}}{\dfrac{i_{OUT}}{v_{SS}}} = \frac{B g_{m1}}{(g_{o8} + 2\pi f C_{n7SS})}, \tag{5.4.12}$$

where C_{n7SS} is the parasitic coupling capacitance between the output node 7 and the supply voltage V_{SS} line.

9. Noise

The equivalent input noise ($\overline{dv_{nie}^2}$) is given by the following expression [46]

$$\overline{dv_{nic}^2} = 2\left[\overline{dv_{n1}^2} + \overline{dv_{n4}^2}\left(\frac{g_{m4}}{g_{m1}}\right)^2 + \frac{1}{B^2}\left\{\left(\frac{g_{m6}}{g_{m1}}\right)^2 \overline{dv_{n6}^2} + \left(\frac{g_{m7}}{g_{m1}}\right)^2 \overline{dv_{n7}^2}\right\}\right]. \qquad (5.4.13)$$

where $\overline{dv_{n1}^2}$, $\overline{dv_{n4}^2}$, $\overline{dv_{n6}^2}$, and $\overline{dv_{n7}^2}$ are the equivalent noise due to M1, M4, M6, and M7 referred to their corresponding input terminals respectively.

10. Power dissipation (PD)

The total power dissipated (PD) is given by

$$PD = (V_{DD} - V_{SS}) \times 2(I_{bias} + I_{D5}). \qquad (5.4.14)$$

11. Area

The total area is obtained by taking the sum of the active areas taken by the individual transistors and is given by

$$AREA = \sum_{i=1}^{10} W_i \times L_i. \qquad (5.4.15)$$

The initial values of the design variables are obtained using the following algorithm.

1. The value of the bias current I_{bias} is obtained from the specified values of the slew rate and the load capacitance C_L, by choosing a suitable value of the current multiplication factor B, and using Eqn.(5.4.4).

2. Assuming a typical value for $(V_{GS} - V_t)_1$, the aspect ratio of M1 is obtained, and, subsequently, the value of its transconductance (g_{m1}) is determined.

3. The aspect ratios of other transistors are determined by assuming suitable values for their respective ($V_{GS} - V_t$). The output resistance of the circuit is obtained from the specified value of the gain, the value of B assumed previously, and the calculated value of g_{m1}, by using Eqn.(5.4.1).

4. The lengths of M6 and M8 are taken to be equal, and, therefore, it can be determined using the value of the channel length modulation parameter and that of the total output resistance, obtained from the previous step.

5. The values of the lengths of M1 and M2 are assumed to be equal to that of the minimum channel length, and, hence, their widths are obtained from the corresponding values of their aspect ratios, determined earlier. The values of the widths of the other transistors are also obtained accordingly.

Once the initial values of the circuit parameters are obtained, the computation of the DC operating point and the small signal parameters are performed using the steps mentioned earlier in this chapter. The various performances of the circuit are determined during each step of the iterations using Eqns.(5.4.1–5.4.15). The output results obtained for a set of performance specifications from the design of the symmetrical OTA along with those obtained from SPICE simulation are listed in Table 5.4.1. All the constant parameters used for the design are the same as those for the simple OTA, as described in Subsection 5.2. The design variables obtained from the optimization program and subsequently used for SPICE simulation are presented in Table 5.4.2. The SPICE simulation results of the variation of the gain and the phase of the circuit as a function of frequency with the parameters obtained from the design are plotted in Figs.5.4.2 and 5.4.3 respectively. The values of the gain, unity-gain frequency, and phase margin obtained from the plots are listed in Table 5.4.1.

Table 5.4.1

The output results obtained from the optimization of the design of the symmetrical OTA for a particular set of performance specifications along with those obtained from SPICE simulation.

Sl. No.	*Symbol*	*Type*	*Weight*	*Target value*	*Value obtained from design*	*Value obtained from SPICE simulation*
1.	*Gain* (dB)	≥	0.1	60	61.637	60.62
2.	*UGF* (MHz)	≥	0.1	20	30.125	29.0
3.	*PM* (degree)	≥	0.05	50	67.88	62.0
4.	*SR* (V/μs)	≥	0.1	5	5.002	5.35
5.	*CMR+* (V)	≥	0.05	3.0	3.71	3.835
6.	*CMR−* (V)	≤	0.05	−3.0	-3.11	-3.16
7.	V_{outmax} (V)	≥	0.05	3.5	3.71	3.835
8.	V_{outmin} (V)	≤	0.05	-3.5	-3.71	3.835
9.	*CMRR* (dB)	≥	0.1	60	60.65	79.37
10.	$PSRR_{DD}$ (dB)	≥	0.05	50	69.189	72.337
11.	$PSRR_{SS}$ (dB)	≥	0.05	50	64.412	65.21
12.	*RMS noise* (nV/√Hz)	≤	0.05	10	4.166	*
13.	*PD* (mW)	≤	0.1	5	2.001	2.14
14.	*AREA* (μ^2)	≤	0.1	200	200.03	*

*SPICE3 in the computer center does not support calculation of these parameters.

Table 5.4.2

The design variables obtained from the optimization routine for the design of the symmetrical OTA, and subsequently used for SPICE simulation.

Sl. No.	*Design variable*	*Obtained value*
1.	W/L of M1, M2 (μm/μm)	97.442/0.8
2.	W/L of M3, M4 (μm/μm)	1.605/3.152
3.	W/L of M5 (μm/μm)	2.991/3.152
4.	W/L of M6 (μm/μm)	3.098/3.152
5.	W/L of M7 (μm/μm)	1.426/3.152
6.	W/L of M8 (μm/μm)	1.329/3.152
7.	W/L of M9, M10 (μm/μm)	1.652/1.849
8.	Bias current (I_{bias}) (μA)	50.2

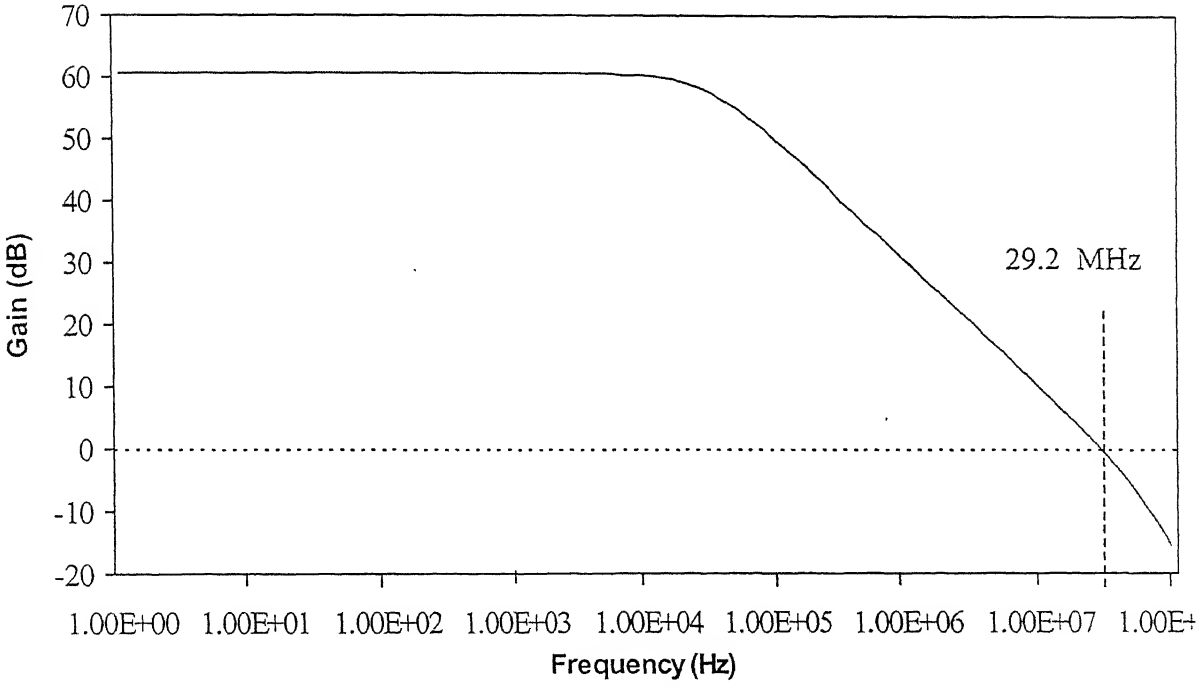


Fig.5.4.2: The gain versus frequency plot for the symmetrical CMOS OTA as obtained from SPICE simulation, using the optimized values of the design variables satisfying the set of performance specifications as shown in Table 5.4.1. The unity-gain frequency of 29.2 MHz is also shown in the figure.

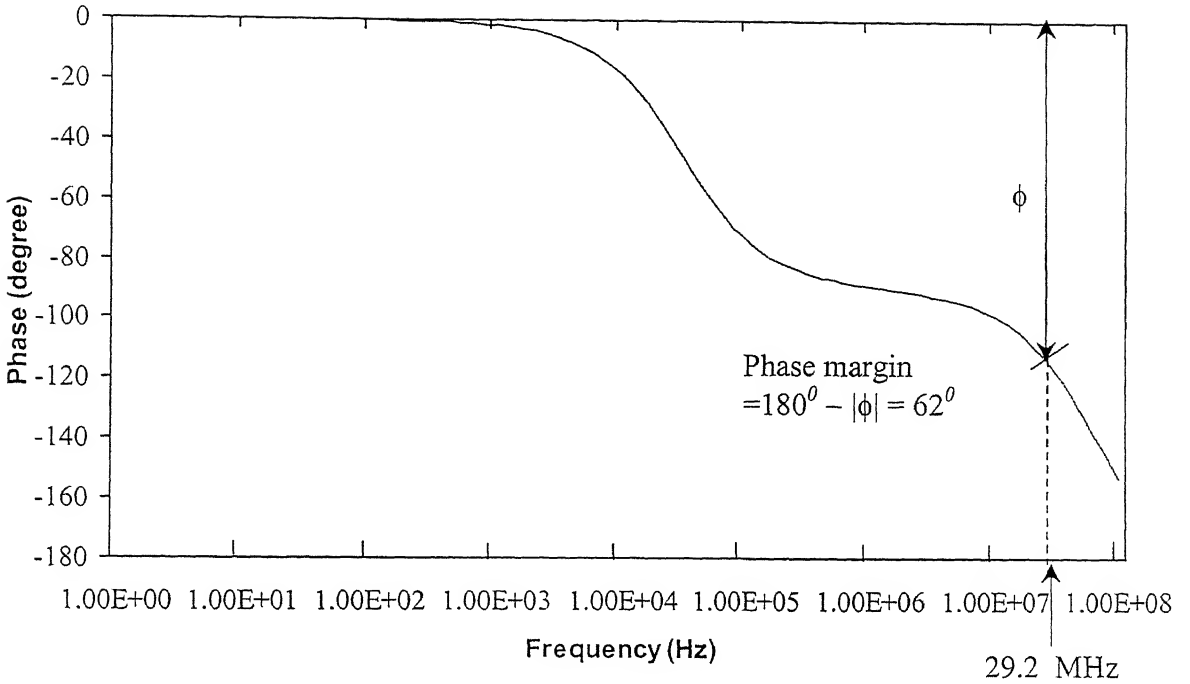


Fig.5.4.3: The phase versus frequency plot for the symmetrical CMOS OTA obtained from SPICE simulation using the optimized values of the design variables satisfying the set of performance specifications as given in Table 5.4.1. The phase margin of 62^0 at the unity-gain frequency of 29.2 MHz is also shown in the figure.

The values of the performance specifications predicted by our algorithm closely match with those obtained from SPICE simulation. However, the higher value of the unity-gain frequency is obtained at the expense of the current drawn from the supply, which is at least B times that of the simple OTA. The higher value of the current leads to a larger power dissipation. The current through the input transistors and their drain-source voltages are equal due to the symmetrical loading in this type of OTA. This ensures that their small signal parameters are the same, and, thus, the transconductance mismatch is avoided, which is an advantage for this type of op-amp [46]. The current mirrors in the topology can be substituted by more complicated ones. The active area taken by this op-amp generally has larger value than the previous two cases, since it incorporates more number of transistors than the other two.

CHAPTER 6

SUMMARY AND CONCLUSION

In this work, we have made an attempt to bring about the recent trend in analog DA, the problems associated with it, and the present state of the art. The analog DA over the years is found to be gaining popularity as a potential research area. Because of the growing requirements of the mixed-signal circuits in a variety of applications, e.g., telecommunications, robotics, etc., and their complexity, the development of computer-aided design (CAD) tools has become indispensable. Over the years, the development of analog CAD tools has not proceeded apace of the digital tool development because of the problems associated with the design of analog circuits. In brief, the design of analog circuits is much more knowledge intensive than its digital counterparts. The goal of the analog synthesis process is to generate a functional circuit design from the target performance specifications, i.e., one should be able to obtain a physical layout of the circuit that meets the specifications. The transformation of the circuit to its layout is a routine task, and automated tools are available, e.g., LASI [41], etc. However, the computation of the values of the circuit parameters that would satisfy a large number of performance requirements for a particular topology is not an easy and obvious task.

For the parametric optimization of analog circuits, analytical equation based constrained optimization method is the most promising approach. Unfortunately, most of the optimization techniques applied for synthesis of analog circuits as reported in the literature are very much rigid in terms of capturing the words expressed by users in order to specify the objectives, and are often difficult to adapt to the design problem without a loss of accuracy. In order to model the real world terms, e.g., *high, maximize, minimize,* etc., used by the user while specifying the performance objectives, we have applied the concept of fuzzy membership functions [39]. A library is developed for different

membership functions, which are subsequently used by the modules for synthesis of different analog circuits. Among a set of performance specifications, some are considered as objectives and others as constraints, and using suitable values of the targets and the tolerance limits, the corresponding membership functions are formulated. All the performance specifications, thus being classified as either fuzzy objectives or fuzzy constraints or strict constraints, are clubbed together by using suitable values of weights assigned to each of them. The consideration that the maximum sum of the weights assigned is equal to unity ensures that the designer, at the end of each of the iterations, precisely knows the distance of the optimal point (i.e., unity) from that particular point. In this work, the solution to the objective function is obtained by using Powell's direct search algorithm [38].

For the use of the tool developed in this work, it is assumed that the designer has a fair knowledge of the circuit topology and approximate values of the performance specifications achievable by it. Also, the designer has to decide the relative importance of each of the specifications in the entire set of specifications. In order to develop the optimization modules for other circuit topologies, an extensive circuit knowledge is required. For any change in the circuit topology, the optimization routine has to be suitably modified. Therefore, development of cell libraries using this approach seems to be quite a tedious task.

The work was started with an aim to build a synthesis tool for the design of analog circuits. We have implemented a new approach as stated in FPAD [33] for designing simple blocks in bipolar as well as in MOS technology. The basic bipolar circuits for which optimization modules are developed in this work are the common-emitter stage as an amplifier, the common-base stage as an amplifier, and the emitter-follower as an output stage. A wide variety of current sources (e.g., simple, cascode, Wilson, modified Wilson, and regulated cascode) are optimized using the tool developed in this work. The synthesis procedure to optimize the common-source amplifier as a gain stage has also been automated. The tool currently supports the optimization of the design procedures for three basic CMOS op-amp topologies, i.e., the simple operational transconductance amplifier

(OTA), the symmetrical OTA, and the basic two-stage (BTS) op-amp. However, using the methodology adopted in this work, a wide variety of other op-amp topologies can also be synthesized. A library of basic analog circuit building blocks can be maintained by developing optimization routines for other simple topologies, which can be used later for the design of more complex circuits.

In order to explore the viability of the tool developed, all the topologies mentioned above are designed for the given set of performance specifications supported by it. Once optimization of the design is done, the program creates three separate files containing the design variables and the output results, the operating point information and the small signal parameters for the individual transistors, and a netlist of the circuit, which can be directly used for SPICE simulation. Before creating each of the above files, the user is asked about the information he wants and the program accordingly stores this information in a file. These information stored during the process help the designer to verify the validity of the design by comparing the results obtained from SPICE simulation with those obtained from the optimization algorithm. The performances predicated by the algorithm for the blocks designed using the tool match reasonably well with those obtained from SPICE simulation. In particular for MOS circuits, the channel length modulation parameter (λ) is taken into consideration for the computation of the DC operating point and the small signal parameters, which has been neglected by most of the earlier approaches.

However, the following modifications and additions are required in order to get a complete tool, which can be easily handled by an inexperienced designer.

- Development of modules for other simple blocks in order to provide a wide variety of choices to the user.
- Development of a graphical user interface (GUI), which can be used to read inputs supplied by the user and display the output results directly.
- Interfacing the circuit simulation tool, e.g., SPICE, that will enable the designer to simulate and verify the design simultaneously.

- Development of routines for a group of similar circuit topologies in order to select the best possible one among them (known as topology selection) and then design using the tool developed in this work.

- A more efficient optimization algorithm, e.g., Genetic Algorithm (GA) [30,31] could be implemented to solve the optimization problem in order to get the global optimum solution.

- Interfacing of the layout generation tools, e.g., LASI [41], etc., in order to generate the layout of the circuit and subsequently, optimization can be performed taking into account the exact parasitic capacitances obtained from the layout of the circuit.

- In order to implement the design of analog circuits of higher level hierarchy, e.g., the phase-locked loop (PLL), the switched-capacitor filter, etc., a suitable methodology, has to be developed to deduce the specifications of the basic building blocks from the specifications of the more complex analog blocks. Recently developed ARCHGEN [47] presents one of the methodologies for the synthesis of analog systems.

REFERENCES

1. M. Ismail and J. Franca, *Introduction to Analog VLSI Design Automation*, Kluwer Academic Publishers, London, 1990.

2. D. Foty, "Taking a deep look at analog CMOS", *IEEE Circ. Devices*, vol. 15, no. 2, pp. 23–28, March 1999.

3. Y. P. Tsividis, "Analog MOS integrated circuits – Certain new ideas, trends and obstacles", *IEEE J. Solid-State Circuits,* vol. SC–22, no. 3, pp. 317–321, June 1987.

4. C. Toumazou and C. A. Makris, "Analog IC design automation: Part I – Automated circuit generation: New concepts and methods", *IEEE Trans. Computer-Aided Design*, vol. 14, no. 2, pp. 218–238, February 1995.

5. L. R. Carley and R. A. Rutenbar, "How to automate analog IC design", *IEEE Spectrum,* vol. 25, no. 8, pp. 26–30, August 1988.

6. B. A. A. Antao and A. J. Brodersen, "Techniques for synthesis of analog integrated circuits", *IEEE Design & Test of Computers*, vol. 14, no. 2, pp. 8–17, March 1992.

7. J. H. Huijsing, R. J. Plassche, and W. Sansen, *Analog Circuit Design*, Kluwer Academic Publishers, London, 1993.

8. G. Kelson, "Design automation techniques for analog VLSI", *VLSI Design*, vol. 5, no. 1, pp. 78–82, January 1985.

9. D. C. Stone, J. E. Schroder, R. H. Kaplan, and A. R. Smith, "Analog CMOS building blocks for custom and semicustom applications", *IEEE J. Solid-State Circuits,* vol. SC–19, no. 1, pp. 56–61, February 1984.

10. T. Pletersek, J. Trontelj, L. Trontelj, I. Jones, and G. Shenton, "High performance designs with CMOS analog standard cells", *IEEE J. Solid-State Circuits,* vol. SC–21, no. 2, pp. 215–222, April 1986.

11. M. J. S. Smith, C. Portmann, C. Anagnostopoulis, P. S. Tschang, P. V. R. Rao, and H. Ching, "Cell libraries and assembly tools for analog/digital CMOS/BiCMOS application specific integrated circuit design," *IEEE J. Solid-State Circuits,* vol. SC–24, no. 5, pp. 1419–1432, December 1989.

12. R. Harjani, R. A. Rutenbar, and L. R. Carley, "OASYS: A framework for analog circuit synthesis", *IEEE Trans. Computer-Aided Design*, vol. 8, no. 6, pp. 680–691, June 1990.

13. F. El-Turkey and E. E. Perry, "BLADES: An artificial intelligence approach to analog circuit design", *IEEE Trans. Computer-Aided Design*, vol. 8, no. 12, pp. 1247–1265, December 1989.

14. E. Berkcan, M. Abreu, and W. Laughton, "Analog compilation based on successive decompositions", in *Proc. ACM/IEEE Design Automat. Conf. (DAC)*, pp. 369–375, 1988.

15. M. R. Degrauwe, O. Nys, and H. J. Oguey, "IDAC: An interactive design tool for analog CMOS circuits", *IEEE J. Solid-State Circuits*, vol. SC–22, no. 6, pp. 1106–1116, December 1987.

16. H. Y. Koh, C. H. Sequin, and P. R. Gray, "OPASYN: A compiler for CMOS operational amplifiers", *IEEE Trans. Computer-Aided Design*, vol. 9, no. 2, pp. 113–125, February 1990.

17. H. Odonera, H. Kanbara, and K. Tamaru, "Operational-amplifier compilation with performance optimization", *IEEE J. Solid-State Circuits*, vol. 25, no. 2, pp. 466–473, April 1990.

18. G. G. E. Gielen, H.C.C. Walscharts, and W. M. C Sansen, "Analog circuit design optimization based on symbolic simulation and simulated annealing", *IEEE J. Solid-State Circuits*, vol. SC–25, no. 3, pp. 707–713, June 1990.

19. G. G. E. Gielen, H. C. C. Walscharts, and W. M. C. Sansen, "ISSAC: A symbolic simulator for analog integrated circuits", *IEEE J. of Solid-State Circuits*, vol. SC–24, no. 6, pp. 1587–1596, December 1989.

20. A. H. Fung, B. W. Lee, and B. J. Sheu, "Self reconstructing technique for expert system-based analog IC designs", *IEEE Trans. Circuits Syst.*, vol. CAS–36, no. 2, pp. 318–321, February 1989.

21. B. J. Sheu, A. H. Fung, and Y. Y. Lai, "A knowledge-based approach to analog IC design", *IEEE Trans. Circuits Syst.*, vol. CAS–35, no. 2, pp. 256–258, February 1988.

22. C. Toumazou and C. A. Makris, "Analog IC design automation: Part II – Automated circuit correction by qualitative reasoning", *IEEE Trans. Computer-Aided Design*, vol. 14, no. 2, pp. 239–254, February 1995.

23. R. K. Brayton, G. D. Hatchel, and A. L. Sangio-Vincentelli, "A survey of optimization techniques for integrated-circuit design", in *Proc. IEEE*, vol. 69, no. 10, pp. 1334–1362, October 1981.

24. M. R. Lightner and S. W. Director, "Multiple criterion optimization for design of electronic circuits", *IEEE Trans. Circuits Syst.*, vol. CAS–28, no. 3, pp. 169–179, March 1981.

25. W. Nye, D. C. Riley, A. Sangiovanni, and A. L. Tits, "DELIGHT.SPICE: An optimization-based system for the design of integrated circuits", *IEEE Trans. Computer-Aided Design*, vol. 7, no. 4, pp. 501–519, April 1988.

26. J. M. Shyu and A. Sangiovanni-Vincetelli, "ECSTASY: A new environment for IC design optimization", in *Proc. IEEE Inter. Conf. Computer-Aided Design (ICCAD)*, pp. 484–487, 1988.

27. J. C. Lai, J. S. Kueng, H. J. Chen, and F. J. Fermandez, "ADOPT – A CAD tool for analog circuit design", *IEEE Circ. Devices*, vol. 18, no. 2, pp. 29–30, March 1988.

28. E. S. Ochotta, R. A. Rutenbar, and L. R. Carley, "Synthesis of high performance analog circuits in ASTRX/OBLX", *IEEE Trans. Computer-Aided Design*, vol. 15, no. 3, pp. 273–293, March 1996.

29. W. H. Press, *Numerical Recipes in C–The Art of Scientific Computing*, Cambridge Univ. Press, Cambridge, U. K., 1988.

30. D. E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*, Addison-Wesley, Massachusetts, 1989.

31. M. Gen and R. Chang, *Genetic Algorithms and Engineering Design*, John Wiley & Sons, New York, 1997.

32. J. P. Harvey, M. I. Elmasry, and B. Leung, "STAIC: An interactive framework for synthesizing CMOS and BiCMOS analog circuits", *IEEE Trans. Computer-Aided Design*, vol. 11, no. 11, pp. 1402-1417, November 1992.

33. M. Fares and B. Kaminska, "FPAD: A fuzzy nonlinear programming approach to analog circuit design", *IEEE Trans. Computer-Aided Design*, vol. 14, no. 7, pp. 785–793, July 1995.

34. A. Torralba, J. Chavez, and L.G. Franquelo, "FASY: A fuzzy logic based tool for analog synthesis", *IEEE Trans. Computer-Aided Design*, vol. 15, no. 7, pp. 785–793, July 1996.

35. P. Mandal and V. Viswanathan, "A new approach for CMOS Op-Amp Synthesis", *12th International Conference on VLSI Design*, pp. 189–194, January 1999.

36. P. C. Maulik, L. R. Carley, and D. J. Allstot, "Sizing of cell level analog circuits using constrained optimization techniques", *IEEE J. Solid-State Circuits*, vol. SC–28, no. 3, pp. 233–241, March 1993.

37. A. F. Schwarz, *Computer-Aided Design of Microelectronic Circuits and Systems, vol. 1, General introduction and Analog circuit aspects*, Academic Press, London, 1987.

38. S. S. Rao, *Optimization Theory & Applications*, 2nd Ed., Wiley Eastern Ltd., New Delhi, 1984.

39. L. A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes", *IEEE Trans. Syst. Man Cybernetics*, vol. SMC–3, no. 1, pp. 28–44, January 1973.

40. K. Deb, *Optimization for Engineering Design*, Prentice Hall India (Pvt.) Ltd., New Delhi, 1995.

41. R. J. Baker, H. L. Wi, and D. E. Boyce, *CMOS: Circuit design, layout and simulation,* IEEE Press, The IEEE Inc., New York, 1998.

42. P. R. Gray and R. G. Meyer, *Analysis and Design of Analog Integrated Circuits*, 3rd Edition, John Wiley and Sons, New York, 1993.

43. R. Gregorian and G. C. Temes, *Analog MOS Integrated Circuits for Signal Processing*, Wiley and Sons, New York, 1984.

44. R. L. Geiger, P. E. Allen, and N. R. Strader, *VLSI Design Techniques for Analog and Digital Circuits*, McGraw-Hill, New York, 1990.

45. P. R. Gray and R. G. Meyer, "MOS operational amplifer design – A tutorial overview", *IEEE J. Solid-State Circuits*, vol. SC–17, no. 6, pp. 969-982, December 1982.

46. K. R. Laker and W. M. C. Sansen, *Design of Analog Integrated Circuits and Systems*, McGraw-Hill, New Jersey, 1994.

47. B. A. A. Antao and A. J. Brodersen, "ARCHGEN: Automated synthesis of analog systems", *IEEE Trans. VLSI Systems*, vol. 3, no. 2, pp. 231–244, June 1995.

APPENDIX-I

THE OPTIMIZATION ALGORITHM

As mentioned earlier in Subsection 2.3, the formulated problem (i.e., the objective function) can be solved with the help of any standard optimization algorithm. Powell's direct search algorithm [38] is one of the widely accepted optimization techniques used for solving unconstrained optimization problems. We have implemented this method because of the simplicity of the algorithm for the development of the computer code. However, the optimization problem considered in this work is a constrained one, and therefore to apply Powell's algorithm for the solution to this problem, it is converted to an unconstrained one. This conversion can be performed by using one of the many transformation techniques [38], e.g., the method of change of variables, the penalty function method, etc. In this work, the transformation is performed using the appropriate change of variables method as illustrated in [38].

In the given problem, a constraint on the design variable vector X_i, which is bound by an upper limit and a lower limit, would always be encountered, which can be mathematically represented as follows

$$L_i \leq X_i \leq U_i, \tag{A1.1}$$

where L_i and U_i are the lower and the upper limits on the design vector X_i respectively. These types of constraints can be satisfied automatically by the transformation of the variable X_i as per the following equation [38]

$$X_i = L_i + (U_i - L_i) \sin^2 (Y_i), \tag{A1.2}$$

where Y_i is the new variable which can take any value.

The algorithm described in this section explains the minimization procedure of a given objective function. It is to be noted that by using the negative of a given function as the objective function in the minimization routine, one can always find the maximum value of the function, and this technique is followed in this work. In order to understand the basic idea involved in Powell's direct search algorithm in order to optimize a given function, a function with two variables x_1 and x_2 is considered. Figure 2.6.1 shows the progress of the algorithm in the variable space, starting from the initial design variables as supplied by the user or obtained from the first cut algorithm developed in this work. The figure shows the two-dimensional floor, along with the contours associated with the function, as it is plotted in a three-dimensional space. The contours can be of any arbitrary shape, and entirely depend upon the nature of the function.

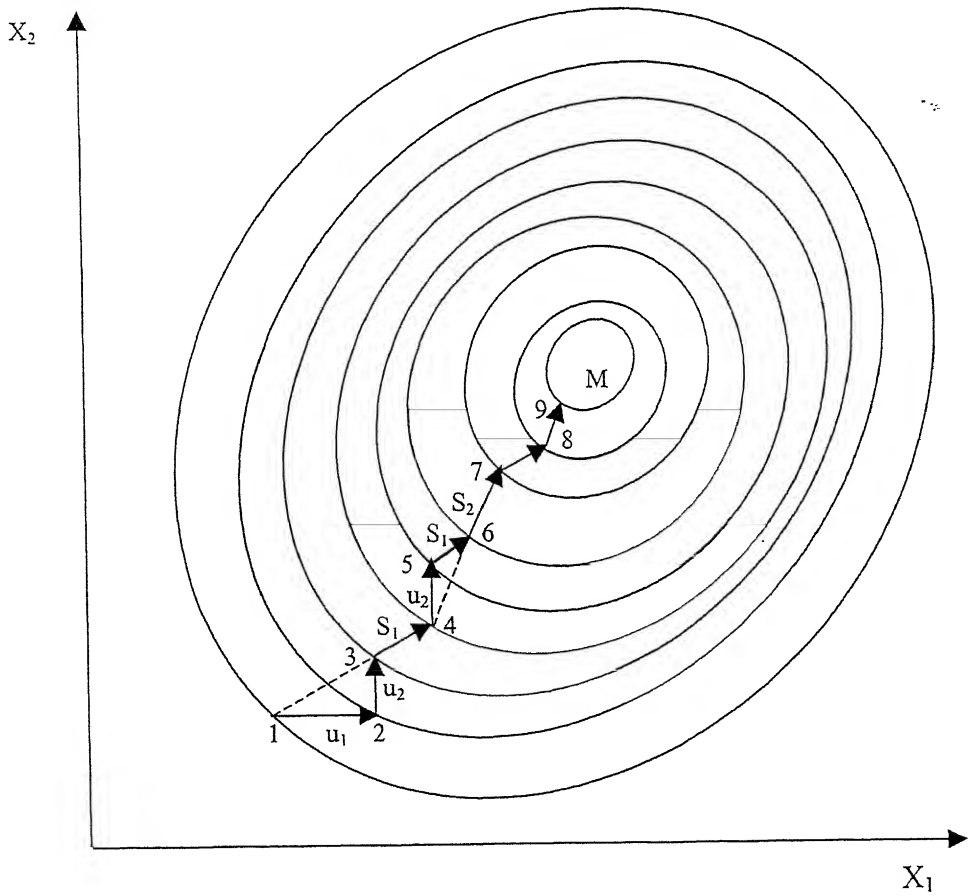


Fig. A1.1: The progress of Powell's direct search algorithm for a two-variable function, starting from the initial point 1 and terminating at point 9 (taken from [38]).

The contours for a given function are considered to be of elliptical shape, and the minimum is assumed to be located at the contour labeled M, as shown in Fig.A1.1. The value of the function decreases from unity to the minimum at point M, with the intermediate contours corresponding to decreasing values of the function. Let the initial values of the design variables are x_{10} and x_{20} correspond to point **1**. The function is initially minimized along the coordinate axis X_1 along the direction vector u_1 from point **1** to **2.** The value of the variable at point **2** is obtained from the addition of the vector u_1 to the initial point **1.** In the next step, the function is minimized along X_2 stating from point **2**, which gives point **3** and the direction vector u_2, as shown in the figure. The function is then minimized along the pattern direction $S_1(u_1,u_2)$, thus obtaining point **4.** Thus, at each step, the multivariable optimization problem is boiling down to a single variable one, i.e., the direction along which it is optimized. The single variable optimization is solved by the golden search technique[1].

For the next cycle of minimization, one of the coordinate directions (X_1 in the present case) is discarded in favor of the pattern direction S_1. Thus, minimizing the function along u_2 (point **5** is obtained after minimization), and then along S_1, point **6** is obtained. The new pattern direction S_2 is generated from the two previous directions, u_2 and S_1, and point **7** is obtained by minimizing the function along S_2. In the next cycle, one of the previously used coordinate directions (X_2 in this case) is discarded in favor of the newly generated pattern direction S_2. Then starting from point **7**, minimization is done along directions S_1 and S_2, thereby obtaining points **8** and **9** respectively. This procedure is repeated until the minimum point of the function is obtained. The condition for the termination of the loop is at the point where the difference between two successive computations of the objective function yields a value less than a suitably chosen small number (of the order of 10^{-15} in our program), i.e., in other words, the search has reached a point where the function does not change appreciably around it. Therefore, the algorithm has the limitation of ending with a local optimal solution, even when a global optimum point may exist in the overall design space.

[1] The golden search algorithm is one of the most versatile algorithms used for optimization of single variable problems. The details can be found in [38,39].

The flow chart of Powell's direct search algorithm for an **n**-variable objective function is presented in Fig.A1.2.

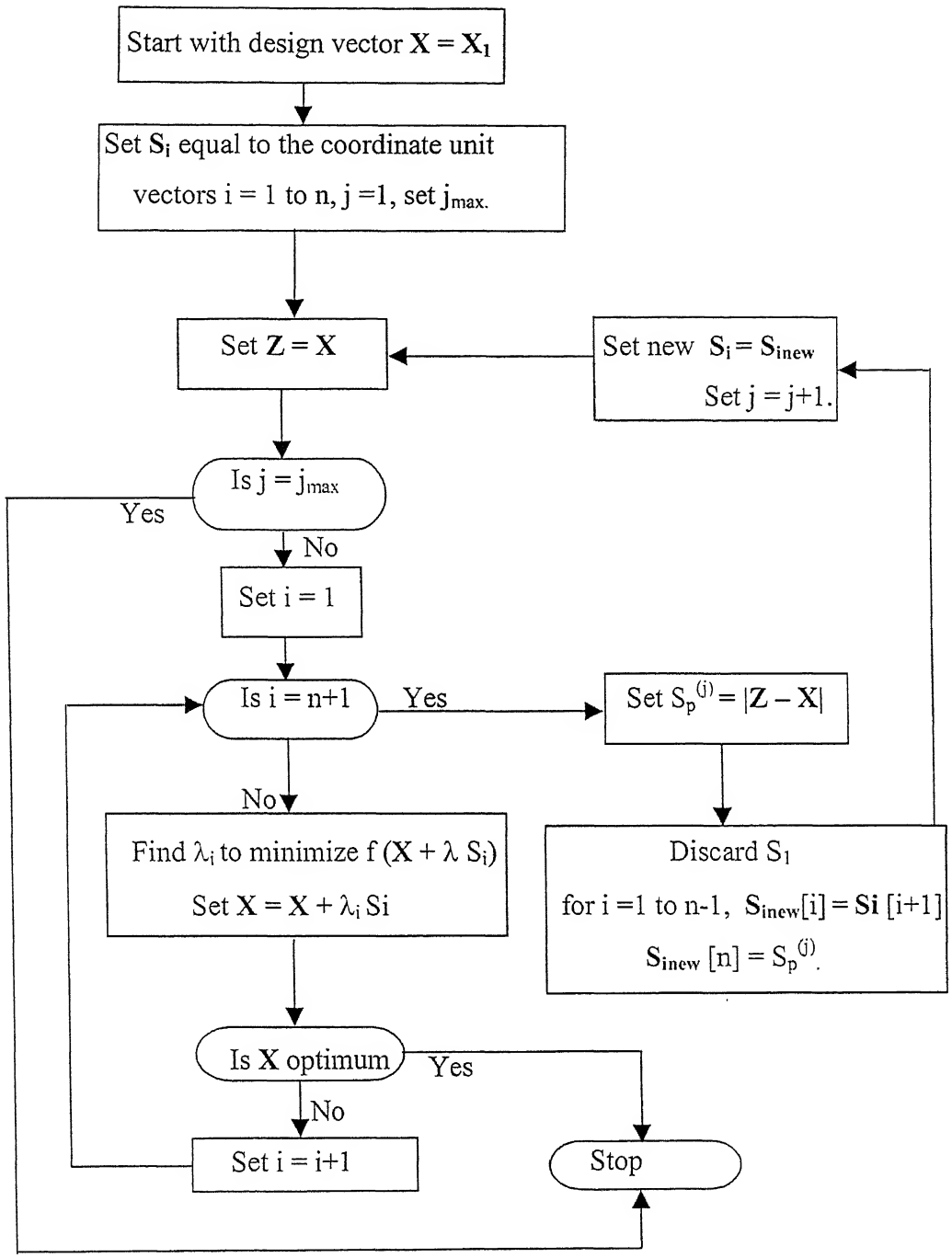


Fig. A1.2: A flow chart representation of Powell's direct search algorithm for optimization of an **n**-variable objective function (taken from [38]).

The algorithm starts with the initial vector of n-variables $\mathbf{X_1}$ (X_1, X_2, ... , X_n). The initial direction vector S_i along which the function is minimized is set to unit vectors along the coordinate axes, i.e., along S_1, S_2,. . . . , and S_n. The objective function is minimized along S_i in the first iteration (for i = 1). During minimization along the direction S_1, a parameter λ_1 is obtained, which indicates the distance of the obtained minimum point from the initial point in terms of the multiples of the unit distance along the search direction. Thus, the new point $\mathbf{X} = \mathbf{X_1} + \lambda_1 S_1$ is obtained. Subsequently, the function is minimized in the loop for i = 2 to n, along the other coordinate directions as presented in the flowchart. If during the process of obtaining the minimum point along these directions, the termination criterion is satisfied, then the program gives the output results with the variables at that point as the variables for the minimum point.

The vector of the initial variables is stored as the previous set of variables, as shown in the flow chart with a separate vector of variables $\mathbf{Z}$ (= $\mathbf{X}$) known as the base point [38]. If the minimum is not obtained even after the algorithm searches along the initial direction set S_i, a pattern direction $S_p^{(j)}$ (= $|\mathbf{Z} - \mathbf{X}|$) is generated. In the next step, the direction vector S_1 is discarded and consequently, the function is minimized along the new set of directions S_i (i.e., along S_2, S_3, S_4, S_n, and $S_p^{(1)}$). In the next iteration, the algorithm searches along the set of directions S_2, S_3, ..., S_{n-1}, S_n, $S_p^{(1)}$, $S_p^{(2)}$, and so on, and the process continues for higher values of the iterations. The search process ends when either the termination criterion for the optimal variable $\mathbf{X}$ is satisfied or the number of iterations (j) exceeds the limit of the maximum number of iterations (j_{max}).

The termination criteria for the present work are twofold. At any point of time during the search process in the design space, if the minimized function attains a value of −1, i.e., the objective function reaches a value of unity (indicating the fulfillment of all the specifications as desired by the user), the program terminates. On the other hand, the program may terminate at a possible minimum point having function value greater than −1, i.e., the objective function value is less than unity, which states that some of the specified performances are not fully met. A detailed explanation of the termination criteria of Powell's algorithm can be obtained in [38,39].

APPENDIX – II

THE BJT TECHNOLOGY FILE

Table A2.1

A typical set of parameters for a high-voltage integrated *npn* transistor with 500 μ^2 emitter area, 5 Ω-cm, 17 μ epi, 44 V device, which is used for designing the various BJT topologies in this work (taken from [42]).

Sl. No.	*Parameter*	*Symbol*	*Value*
1.	Transistor saturation current (fA)	I_S	5
2.	Maximum forward current gain	β_F	200
3.	Maximum reverse current gain	β_R	2
4.	Forward early voltage (V)	V_A	130
5.	Base series resistance (Ω)	r_b	200
6.	Emitter series resistance (Ω)	r_e	2
7.	Collector series resistance (Ω)	r_c	200
8.	Forward Transit time (ns)	τ_F	0.35
9.	Reverse Transit time (ns)	τ_R	400
10.	Zero bias base-emitter capacitance (pF)	C_{JE}	1
11.	Base-emitter junction built-in potential (V)	V_{JE}	0.7
12.	Base-emitter junction capacitance exponent	M_{JE}	0.33
13.	Zero bias base-collector depletion capacitance (pF)	C_{JC}	0.3
14.	Base-collector junction built-in potential (V)	V_{JC}	0.55
15.	Base-collector junction capacitance exponent	M_{JC}	0.5
16.	Zero bias collector-substrate depletion capacitance (pF)	C_{JS}	3
17.	Collector-substrate junction built-in potential (V)	V_{JS}	0.52
18.	Collector-substrate junction capacitance exponent	M_{JS}	0.5
19.	Collector-emitter saturation potential (V)	V_{cesat}	0.2

APPENDIX–III

THE MOS TECHNOLOGY FILE

Table A3.1

A set of process parameters for a typical silicon-gate n-well CMOS process with 0.8 μ minimum allowed gate length, which is used in the design of all MOS circuit topologies in this work (taken from [42]).

Sl. No.	*Parameter*	*Symbol*	*Value*	
			n-channel transistor	*p-channel transistor*
1.	Substrate doping (atoms/cm^3)	N_A, N_D	4×10^{15}	3×10^{16}
2.	Gate oxide thickness (Å)	t_{ox}	150	150
3.	Metal-silicon work function (V)	ϕ_{ms}	-0.6	-0.1
4.	Channel mobility (cm^2/V-sec)	μ_n, μ_p	550	250
5.	Minimum drawn channel length (μ)	L_{drawn}	0.8	0.8
6.	Source, drain junction depth (μ)	X_j	0.2	0.3
7.	Source, drain side diffusion (μ)	L_d	0.12	0.18
8.	Overlap capacitance per unit gate width (fF/μ)	C_{ol}	0.12	0.18
9.	Threshold voltage adjust effective depth (μ)	X_i	0.2	0.2
10.	Threshold voltage adjust effective surface concentration (atoms/cm^2)	N_{si}	3×10^{16}	2×10^{16}
11	Nominal threshold voltage (V)	V_t	0.7	-0.7
12.	Poly-silicon gate doping concentration (atoms/cm^3)	N_{dpoly}	10^{20}	10^{20}
13.	Poly gate sheet resistance (Ω/□)	R_S	10	10

14.	Source/drain-bulk junction capacitance per unit source/drain area (fF/μ^2) (zero bias)	C_{j0}	0.18	0.30
15.	Source/drain-bulk junction capacitance exponent	n	0.5	0.5
16.	Source/drain-periphery capacitance per unit source/drain periphery (fF/μ) (zero bias)	C_{jsw0}	1.0	2.2
17.	Source/ drain-periphery capacitance exponent	n	0.5	0.5
18.	Surface-state density (no./cm^2)	N_{ss}	10^{11}	10^{11}
19.	Rate of change of the drain depletion width with the drain-source voltage (μ/V)	dX_d/dV_{DS}	0.08	0.04
20.	Flicker noise coefficient (C^2/cm^2)	KF	4×10^{-31}	10^{-32}
21.	Flicker noise exponent	AF	1	1